AD 700 028

ELEMENTS OF LARGE SCALE MATHEMATICAL PROGRAMMING

A.M. Geoffrion

The Rand Corporation
Santa Monica, California

November 1969

Elements of Large-Scale Mathematical Programming

A. M. Geoffrion

November 1969

R-481-PR

A REPORT PREPARED FOR

UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

90

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this study should not be interpreted as representing the official opinion or policy of the United States Air Force.

Elements of Large-Scale Mathematical Programming

A. M. Geoffrion

November 1969

R-481-PR

A REPORT PREPARED FOR

UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

This study is presented as a competent treatment of the subject, worthy of publication. The Rand Corporation vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the authors.

Published by The RAND Corporation

PREFACE AND SUMMARY

This report is a part of a continuing Rand research effort in the general area of mathematical programming. Increasingly, the practical problems that very large organizations confront are highly structured, with many decision variables and constraints. In the Air Force, problems of long-term program planning and allocating scarce resources are becoming more complex. The obvious importance of such problems, and the intriguing mathematical possibilities for solving them, have led to a voluminous technical literature. Unfortunately, little has been done to distill and unify the essential concepts found in this literature, with the result that the technical development of the field and its practical application have been retarded.

The aim of this study is to identify and develop the concepts central to the optimization of large structural systems, and to attempt an organization of the literature around these concepts. It is hoped that nonspecialists will find the study a coherent introduction to large-scale optimization, and that the specialist will find it a source of new insights and unifying concepts.

The author carried out this work as a consultant to The Rand Corporation and also under the auspices of a Ford Foundation Faculty Research Fellowship and National Science Foundation Grant GP-8740.

An earlier version of this study was published as Working Paper 144 by the Western Management Science Institute of the University of California at Los Angeles and has been used there and at Stanford University as a supplementary text. It is being published in this form to make it readily available to the Air Force and other users.

ACKNOWLEDGMENTS

It is a pleasure to acknowledge the helpful comments of A. Feinberg, B. L. Fox, and C. A. Holloway.

CONTENTS

1. INTRODUCTION

It is widely held that the development of efficient optimization techniques for large structured mathematical programs is of great importance in economic planning, engineering, and management science. A mere glance at the bibliography of this paper will reveal the enormous effort devoted to the subject in recent years. The purpose of this paper is to suggest a unifying framework to help both the specialist and nonspecialist cope with this vast and rapidly growing body of knowledge.

The framework is based on a relative handful of fundamental concepts. They can be classified into two groups: problem manipulations and solution strategies. Problem manipulations are devices for restating a given problem in an alternative form that is apt to be more amenable to solution. The result is often what is referred to in the literature as a "master" problem. Dualization of a linear program is one familiar example of such a device. Section 2 discusses three others: Projection, Inner Linearization, and Outer Linearization. Solution strategies, on the other hand, reduce an optimization problem to a related sequence of simpler optimization problems. This often leads to "subproblems" amenable to solution by specialized methods. The Feasible Directions strategy is a well-known example, and Sec. 3 discusses the Piecewise, Restriction, and Relaxation strategies. The reader is probably already familiar with special cases of most of these concepts, if not with the names used for them here; the new terminology is introduced to emphasize the generality of the ideas involved.

By assembling these and a few other problem manipulations and solution strategies in various patterns, one can rederive the essential aspects of most known large-scale programming algorithms (and even design new ones). Section 4 illustrates this for Benders Decomposition, Dantzig-Wolfe Decomposition, Rosen's Primal Partition Programming method, Takahashi's "local" approach, and a procedure recently devised by the author for nonlinear decomposition.

Although much of the presentation is elementary, for full appreciation the reader will find it necessary to have a working knowledge of the theory and computational methods of linear and nonlinear programming about at the level of a first graduate course in each subject.

1.1 TYPES OF LARGE-SCALE PROBLEMS

It is important to realize that size alone is not the distinguishing attribute of the field of "large-scale programming," but rather size in conjunction with structure. Large-scale programs almost always have distinctive and pervasive structure beyond the usual convexity or linearity properties. The principal focus of large-scale programming is the exploitation of various special structures for theoretical and computational purposes.

There are, of course, many possible types of structure. Among the commonest and most important general types are these: multidivisional, combinatorial, dynamic, and stochastic. Multidivisional problems consist of a collection of interrelated "subsystems" to be optimized.† The subsystems can be, for example, modules of

---

†See, e.g., Aoki 68, Bradley 67, Gould 59, Hass 68, Kornai and Liptak 55, Lasdon and Schoeffler 66, Malinvaud 67, Manne and Markowitz 63, Parikh and Shephard 67, Rosen and Ornea 63, Tcheng 66.

an engineering system, reservoirs in a water resources system, departments or divisions of an organization, production units of an industry, or sectors of an economy. Combinatorial problems typically have a large number of variables because of the numerous possibilities for selecting routes, machine setups, schedules, etc.[†] Problems with dynamic aspects are large because of the need to replicate constraints and variables to account for several time periods.[††] And problems with stochastic or uncertainty aspects are often larger than they would otherwise be in order to account for alternative possible realizations of imperfectly known entities.[†††] A method that successfully exploits one specific structure can usually be adapted to exploit other specific structures of the same general type. Perhaps needless to say, problems are not infrequently encountered which fall simultaneously into two or more of these general categories.

The presence of a large number of variables or constraints can be due not only to the intrinsic nature of a problem as suggested above, but also to the chosen representation of the problem. Sometimes a problem with a few nonlinearities, for example, is expressed as a completely linear program by means of piecewise-linear or tangential linear approximation to the nonlinear functions or sets (cf.

[†]See, e.g., Dantzig 60, Dantzig, Blattner and Rao 67, Dantzig, Fulkerson and Johnson 54, Dantzig and Johnson 64, Ford and Fulkerson 58, Gilmore and Gomory 61, 63, and 65, Glassey 66, Midler and Wollmer 68, Rao and Zionts 68, Appelgren 69.

[††]See, e.g., Charnes and Cooper 55, Dantzig 55b, 59, Dzielinski and Gomory 65, Glassey 68, Rao 68, Robert 63, Rosen 67, Van Slyke and Wets 66, Wagner 57, Wilson 66.

[†††]See, e.g., Dantzig and Madansky 61, El Agizy 67, Van Slyke and Wets 66, Wolfe and Dantzig 62.

Secs. 2.2, 2.3). Such approximations usually greatly enlarge the size of the problem.[†]

1.2 SCOPE OF DISCUSSION AND THE LITERATURE

The literature on the computational aspects of large-scale mathematical programming can be roughly dichotomized as follows:

I. Work aimed at improving the computational efficiency of a known solution technique (typically the Simplex Method) for special types of problems.

II. Work aimed at developing fundamentally new solution techniques.

The highly specialized nature of the category I literature and the availability of several excellent surveys thereon leave little choice but to focus this paper primarily on category II. Fortunately this emphasis would be appropriate anyway, since category II is far more amorphous and in need of clarification.

Category I

The predominant context for category I contributions is the Simplex Method for linear programming. The objective is to find, for various special classes of problems, ways of performing each Simplex iteration in less time or using less primary storage. This work is in the tradition of the early and successful specialization of the Simplex Method for transportation problems and problems with upper-bounded variables. The two main approaches may be called *inverse compactification* and *mechanized pricing*.

[†] See, e.g., Charnes and Lemke 54, Gomory and Hu 62, Kelley 60.

Inverse compactification schemes involve maintaining the basis inverse matrix or an operationally sufficient substitute in a more advantageous form than the explicit one. One of the earliest and most significant examples is the "product form" of the inverse [Dantzig and Orchard-Hays 54], which takes advantage of the sparseness of most large matrices arising in application. Other schemes involve triangular factorization, partitioning, or use of a "working basis" that is more tractable than the true one. See part A of Table 1. A survey of many such contributions is found in Sec. II of [Dantzig 68]. The interested reader should also consult [Willoughby 69] which, in the course of collecting a number of recent advances in the methods of dealing with sparse matrices, points out much pertinent work done in special application areas such as engineering structures, electrical networks, and electric power systems. Well over a hundred references are given.

Table 1

SOME WORK AIMED AT IMPROVING THE EFFICIENCY OF THE SIMPLEX METHOD FOR LARGE-SCALE PROBLEMS

A. Inverse Compactification
Dantzig and Orchard-Hays 54; Dantzig 55a, 55b, 63b; Markowitz 57; Dantzig, Harvey, and McKnight 64; Heesterman and Sandee 65; Kaul 65; Bakes 66; Bennett 66; Bennett and Green 66; Saigal 66; Dantzig and Van Slyke 67; Sakarovitch and Saigal 67; Grigoriadis 69; Willoughby 69.

B. Mechanized Pricing[a]
Ford and Fulkerson 58; Dantzig 60; Gilmore and Gomory 61,[b] 63, 65; Dantzig and Johnson 64; Bradley 65, Sec. 3; Glassey 66; Tomlin 66; Dantzig, Blattner and Rao 67; Elmaghraby 68; Lasdon and Mackey 68; Rao 68, Sec. II; Rao and Zionts 68; Graves, Hatfield and Whinston 69; Fox 69a.

[a]Most of the references in part C of Table 2 also use mechanized pricing.

[b]Discussed in Sec. 3.2.

Mechanized pricing, sometimes called *column generation*, involves the use of a subsidiary optimization algorithm instead of direct enumeration to find the best nonbasic variable to enter the basis when there are many variables.† The first contribution of this sort was [Ford and Fulkerson 58], in which columns were generated by a network flow algorithm. Subsequent authors have proposed generating columns by other network algorithms, dynamic programming, integer programming, and even by linear programming itself. See part B of Table 1. Excellent surveys of such contributions are [Balinski 64] and [Gomory 63].

Category I contributions of comparable sophistication are relatively rare in the literature on nonlinear problems. It has long been recognized that it is essential to take advantage of the recursive nature of most of the computations; that is, one should obtain the data required at each iteration by economically updating the data available from the previous iteration, rather than by operating each time on the original problem data. In Rosen's gradient projection algorithm, for example, the required projection matrix is updated at each iteration rather than computed ab initio. This is quite different, however, from "compacting" the projection matrix for a particular problem structure, or "mechanizing" the search for the most negative multiplier by means of a subsidiary optimization algorithm. Little has been published along these lines (see, however, p. 153 ff. and Sec. 8.3 of [Fiacco

†It is also possible to mechanize the search for the exiting basic variable when there are many constraints (e.g., Gomory and Hu 62, Sec. 4) or when what amounts to the Dual Method is used (e.g., Sec. 3 of Gomory and Hu 62, Abadie and Williams 63, Whinston 64, and part A of Table 2).

and McCormick 68]). Of course, many nonlinear algorithms involve a sequence of derived linear programs and therefore can benefit from the techniques of large-scale linear programming.

Category II

We turn now to work aimed at developing new solution techniques for various problem structures--the portion of the literature to which our framework of fundamental concepts is primarily addressed.

As mentioned above, the fundamental concepts are of two kinds: problem manipulations and solution strategies. The key problem manipulations (Sec. 2) are Dualization, Projection, Inner Linearization, and Outer Linearization, while the key solution strategies (Sec. 3) are Feasible Directions, Piecewise, Restriction, and Relaxation. These building block concepts can be used to reconstruct many of the existing computational proposals. Using Projection followed by Outer Linearization and Relaxation, for example, we can obtain Benders' Partitioning Procedure. Rosen's Primal Partition Programming algorithm can be obtained by applying Projection and then the Piecewise strategy. Dantzig-Wolfe Decomposition employs Inner Linearization and Restriction. Similarly, many other existing computational proposals for large-scale programming can be formulated as particular patterns of problem manipulations and solution strategies applied to a particular structure.

See Table 2 for a classification of much of the literature of category II in terms of such patterns. One key or representative paper from each pattern is underlined to signify that it is discussed in some detail in Sec. 4. Familiarity with one such paper from each

pattern should enable the reader to assimilate the other papers, given an understanding of the fundamental concepts at the level of Secs. 2 and 3.

Table 2

CLASSIFICATION OF SOME REFERENCES BY PATTERN:
PROBLEM MANIPULATION(S)/SOLUTION STRATEGY

A. Projection, Outer Linearization/Relaxation

Benders 62; Balinski and Wolfe 63; Gomory and Hu 64, pp. 351-354; Buzby, Stone and Taylor 65; Van Slyke and Wets 66, Sec. 2; Weitzman 67; Geoffrion 68b, Sec. 3.

B. Projection/Piecewise

Rosen 63, 64; Rosen and Ornea 63; Beale 63; Gass 66; Varaiya 66; Chandy 68; Geoffrion 68b, Sec. 5; Grigoriadis and Walker 68.

C. Inner Linearization/Restriction

Dantzig and Wolfe 60; Dantzig and Madansky 61, p. 175; Williams 62; Wolfe and Dantzig 62; Dantzig 63a, Ch. 24; Baumol and Fabian 64; Bradley 65, Sec. 2; Dzielinski and Gomory 65; Madge 65; Tcheng 66; Tomlin 66; Whinston 66; Malinvaud 67, Sec. V; Parikh and Shephard 67; Elmaghraby 68; Hass 68; Rao 68, Sec. III; Robers and Ben-Israel 68; Appelgren 69.

D. Projection/Feasible Directions

Zschau 67; Abadie and Sakarovitch 67; Geoffrion 68b, Sec. 4; Silverman 68; Grinold 69, Secs. IV and V.

E. Dualization/Feasible Directions

Uzawa 58; Takahashi 64, "local" approach; Lasdon 64, 68; Falk 65, 67; Golshtein 66; Pearson 66; Wilson 66; Bradley 67 (Sec. 3.2), 68 (Sec. 4); Grinold 69, Sec. III.

Table 2 does not pretend to embrace the whole literature of category II. There undoubtedly are other papers that can naturally be viewed in terms of the five patterns of Table 2, and there certainly

are papers employing other patterns.[†] Sections 2 and 3 mention other papers that can be viewed naturally in terms of one of the problem manipulations or solution strategies discussed there. Still other contributions seem to employ manipulations or strategies other than (and sometimes along with) those identified here;[††] regrettably, this interesting work does not fall entirely within the scope of this effort.

Another group of papers not dealt with in the present study are those dealing with an infinite number of variables or constraints, although a number of contributions along these lines have been made, particularly in the linear case--see, e.g., [Charnes, Cooper and Kortanek 69], [Hopkins 69]. Nor do we consider the literature on mathematical programs in continuous time (a recent contribution with a good bibliography is [Grinold 68]), or literature on the interface between mathematical programming and optimal control theory (e.g., [Dantzig 66], [Rosen 67], [Van Slyke 68]).

1.3 NOTATION

Although the notation employed is not at odds with customary usage, the reader should keep a few conventions in mind.

Lowercase letters are used for scalars, scalar-valued functions, and vectors of variables or constants. Except for gradients (e.g.,

---

[†]E.g.: Inner Linearization/Relaxation: Abadie and Williams 63, Whinston 64.
Dualization, Outer Linearization/Relaxation: Takahashi 64 ("global" approach), Geoffrion 68b (Sec. 6), Fox 69b.
Inner Linearization, Projection, Outer Linearization/Relaxation: Metz, Howard and Williamson 66.
Dualization/Relaxation: Webber and White 68.

[††]E.g.: Balas 65 and 66, Bell 66, Charnes and Cooper 55, Gomory and Hu 62 (Secs. 1 and 2), Kornai and Liptak 65, Kronsjö 68, Orchard-Hays 68 (Ch. 12), Rech 66, Ritter 67b.

$\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}))$, all vectors are column vectors unless transposed. Capital letters are used for matrices (A, B, etc.), sets (X, Y, etc.) and vector-valued functions (e.g., $G(x) = [g_1(x), \ldots, g_m(x)]^t$). The dimension of a matrix or vector-valued function is left unspecified when it is immaterial to the discussion or obvious from context. The dimension of x, however, will always be n. The symbol "$\leqq$" is used for vector inequalities, and "$\leq$" for scalar inequalities. "$\triangleq$" means "equal by definition to." The notation *s.t.*, used in stating a constrained optimization problem, means "subject to." *Convex polytope* refers to the solution set of a finite system of linear equations or inequations; it need not be a bounded set.

2. PROBLEM MANIPULATIONS: SOURCE OF "MASTER" PROBLEMS

A *problem manipulation* is defined to be the restatement of a given problem in an alternative form that is essentially equivalent but more amenable to solution. Nearly all of the so-called *master* problems found in the large-scale programming literature are obtained in this way.

A very simple example of a problem manipulation is the introduction of slack variables in linear programming to convert linear inequality constraints into linear equalities. Another is the restatement of a totally separable problem like (here x_i may be a vector)

$$\underset{x_1,\ldots,x_k}{\text{Minimize}} \sum_{i=1}^{k} f_i(x_i) \quad \text{s.t.} \quad G_i(x_i) \geqq 0,\ i=1,\ldots,k$$

as k independent problems, each of the form

$$\underset{x_i}{\text{Minimize}}\ f_i(x_i) \quad \text{s.t.} \quad G_i(x_i) \geqq 0.$$

This manipulation crops up frequently in large-scale optimization, and will be called *separation*.

These examples, although mathematically trivial, do illustrate the customary purpose of problem manipulation: to permit existing optimization algorithms to be applied where they otherwise could not, or to take advantage in some way of the special structure of a particular problem. The first example permits the classical Simplex Method, which deals directly only with equality constraints, to be applied to linear programs with inequality constraints. The second example enables solving a totally separable problem by the simultaneous solution of smaller

problems. Even if the smaller problems are solved sequentially rather than simultaneously, a net advantage is still probable since for most solution methods the amount of work required increases much faster than linearly with problem size.

More specifically, the three main objectives of problem manipulation in large-scale programming seem to be:

(a) to isolate familiar special structures imbedded in a given problem (so that known efficient algorithms appropriate to these structures can be used);

(b) to induce linearity in a partly nonlinear problem via judicious approximation (so that the powerful linear programming algorithms can be used);

(c) to induce separation.

We shall discuss in detail three potent devices frequently used in pursuit of these objectives: Projection, Inner Linearization, and Outer Linearization.

Projection (Sec. 2.1), sometimes known as "partitioning" or "parameterization", is a device which takes advantage in certain problems of the relative simplicity resulting when certain variables are temporarily fixed in value. In [Benders 62] it is used for objective (a) above to isolate the linear part of a "semilinear" program (see Sec. 4.1), while in [Rosen 64] it is used to induce separation (see Sec. 4.2).

Inner Linearization (Sec. 2.2) and Outer Linearization (Sec. 2.3) are devices for objective (b) long used in nonlinear programming. Inner Linearization goes back at least to [Charnes and Lemke 54], in which a convex function of one variable is approximated by a piecewise-linear convex function. Outer Linearization involves tangential

approximation to convex functions as in [Kelley 60] (see Sec. 3.3). Both devices have important uses in large-scale programming. Inner Linearization is the primary problem manipulation used in the famous Dantzig-Wolfe Decomposition method of linear and nonlinear programming (Sec. 4.3). One important use of Outer Linearization is as a means of dealing with nonlinearities introduced by Projection (Sec. 4.1).

Perhaps the most conspicuous problem manipulation not discussed here is Dualization. Long familiar in the context of linear programs, dualization of nonlinear programs[†] is especially valuable in pursuit of objectives (a) and (c). This significant omission is made because of space considerations, and also to keep the presentation as elementary as possible. One algorithm relying on nonlinear dualization is mentioned in Sec. 4.5; see also part E of Table 2 and [Geoffrion 68b; Sec. 6.1].

Other problem manipulations not discussed here, mostly quite specialized, can be found playing conspicuous roles in [Charnes and Cooper 55], [El Agizy 67], [Gomory and Hu 62], [Weil and Kettler 68].

We now proceed to discuss Projection and Inner and Outer Linearization. Section 3 will discuss the solution strategies that can be applied subsequent to these and other problem manipulations. The distinction between problem manipulations and solution strategies is that the former replaces an optimization problem by one that is essentially equivalent to it, while the latter replaces a problem by a recursive sequence of related but much simpler optimization problems.

[†]See, e.g., Rockafellar 68, Geoffrion 69.

2.1 PROJECTION

The problem

$$(2.1)\qquad \underset{\substack{x\in X\\ y\in Y}}{\text{Maximize}}\ f(x,y)\quad \text{s.t.}\quad G(x,y) \geq 0$$

involves optimization over the joint space of the x and y variables. We define its *projection* onto the space of the y variables alone as

$$(2.2)\qquad \underset{y\in Y}{\text{Maximize}}\ \left[\underset{x\in X}{\text{Sup}}\ f(x,y)\quad \text{s.t.}\quad G(x,y) \geq 0\right].$$

The maximand of (2.2) is the entire bracketed quantity--call it v(y)--which is evaluated, for fixed y, as the supremal value of an "inner" maximization problem in the variables x. We define v(y) to be $-\infty$ if the inner problem is infeasible. The only constraint on y in (2.2) is that it must be in Y, but obviously to be a candidate for the optimal solution y must also be such that the inner problem is feasible, i.e., y must be in the set

$$(2.3)\qquad V \triangleq \{y: v(y) > -\infty\} \equiv \{y: G(x,y) \geq 0 \text{ for some } x\in X\}.$$

Thus we may rewrite (2.2) as

$$(2.4)\qquad \underset{y\in Y\cap V}{\text{Maximize}}\quad v(y).$$

The set V can be thought of as the projection of the constraints $x\in X$ and $G(x,y) \geq 0$ onto the space of the y variables alone. It is depicted for a simple case in Fig. 1; X is an interval, the set

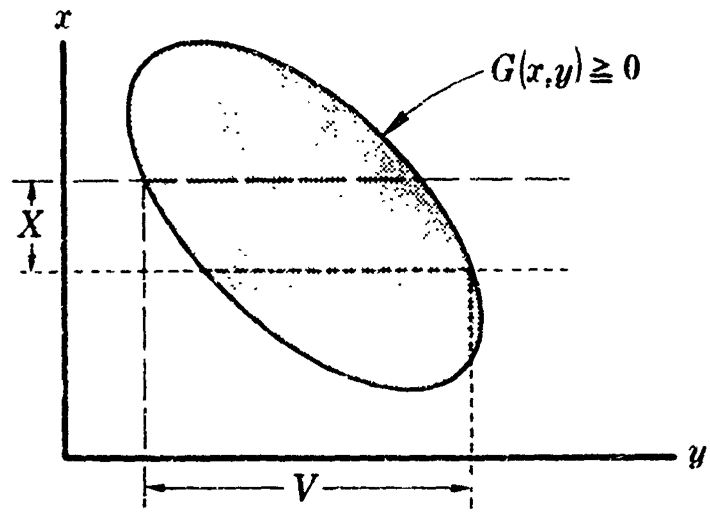

Fig. 1--Depiction of the set V

$\{(x,y):G(x,y) \geqq 0\}$ is shaded, and the resulting V is an interval. It is often possible to obtain a more conventional and tractable representation o than the definitional one. See, for example, the inequalities (4.5) of Sec. 4.1 (cf. [Kohler 67]).

The relationship between the original problem (2.1) and its projection (2.4) is as follows.[†] The proof is elementary.

> <u>Theorem 1</u>. Problem (2.1) is infeasible or has unbounded value if and only if the same is true of (2.4). If (x^o, y^o) is optimal in (2.1), then y^o must be optimal in (2.4). If y^o is optimal in (2.4) and x^o achieves the supremum of $f(x, y^o)$ subject to $x \in X$ and $G(x, y^o) \geqq 0$, then x^o together with y^o is optimal in (2.1).

[†]One may read (2.2) for (2.4) in Theorem 1, except that (2.2) can be feasible with value $-\infty$ when (2.1) is infeasible.

It should be emphasized that Projection is a very general manipulation--no special assumptions on X, Y, f, or G are required for Th. 1 to hold, and any subset of variables whatsoever can be designated to play the role of y. When convexity assumptions do hold, however, the following theorem shows that (2.2) is a concave program.

<u>Theorem 2</u>. Assume that X and Y are convex sets, and that f and each component of G are concave on $X \times Y$. Then v is concave on Y.

<u>Proof</u>. Fix y^o, $y' \epsilon Y$ and $0 < \theta < 1$ arbitrarily. Let $\bar{\theta} = 1 - \theta$. Then

$v(\theta y^o + \bar{\theta} y') =$

$$\operatorname*{Sup}_{x^o, x' \epsilon X} f(\theta x^o + \bar{\theta} x', \theta y^o + \bar{\theta} y') \quad \text{s.t.} \quad G(\theta x^o + \bar{\theta} x', \theta y^o + \bar{\theta} y') \geqq 0$$

$$\geq \operatorname*{Sup}_{x^o, x' \epsilon X} f(\theta x^o + \bar{\theta} x', \theta y^o + \bar{\theta} y') \quad \text{s.t.} \quad G(x^o, y^o) \geqq 0, G(x', y') \geqq 0$$

$$\geq \operatorname*{Sup}_{x^o, x' \epsilon X} \theta f(x^o, y^o) + \bar{\theta} f(x', y') \quad \text{s.t.} \quad G(x^o, y^o) \geqq 0, G(x', y') \geqq 0$$

$= \theta v(y^o) + \bar{\theta} v(y')$,

where the equality or inequality relations follow, respectively, from the convexity of X, the concavity of G, the concavity of f, and separability in x^o and x'. ||

Since V is easily shown to be a convex set when v is concave, it follows under the hypotheses of Theorem 2 that (2.4) is also a concave program.

Projection is likely to be a useful manipulation when a problem is significantly simplified by temporarily fixing the values of certain variables. In [Benders 62], (2.1) is a linear program for fixed y (see Sec. 4.1). In [Rosen 64], (2.1) is a separable linear program for fixed y (see Sec. 4.2). See Table 2 for numerous other instances in which Projection plays an important role.

It is interesting to note that Projection can be applied sequentially by first projecting onto a subset of the variables, then onto a subset of these, and so on. The result is a dynamic-programming-like reformulation [Bellman 57], [Dantzig 59, p. 61 ff.], [Nemhauser 64]. Many dynamic programming problems can fruitfully be viewed in terms of sequential projection, and conversely, but we shall not pursue this matter here.

It may seem that the maximand of the projected problem (2.2) is excessively burdensome to deal with. And indeed it may be, but the solution strategies of Sec. 3 enable many applications of Projection to be accomplished successfully. The key strategies seem to be Relaxation preceded by Outer Linearization (cf. Sec. 4.1), the Piecewise strategy (cf. Sec. 4.2), and Feasible Directions (cf. Sec. 4.4). Of course if y is only one-dimensional, (2.2) can be solved in a parametric fashion [Joksch 64], [Ritter 67a].

2.2 INNER LINEARIZATION

Inner Linearization is an approximation applying both to convex or concave functions and to convex sets. It is conservative in that it does not underestimate (overestimate) the value of a convex (concave) function, or include any points outside of an approximated convex set.

An example of Inner Linearization applied to a convex set X in two dimensions is given in Fig. 2, where X has been approximated by the convex hull of the points $x^1,...,x^5$ lying within it. X has been linearized in the sense that the approximating set is a convex polytope (which, of course, can be specified by a finite number of linear inequalities). The points $x^1,...,x^5$ are called the *base*. The accuracy of the approximation can be made as great as desired by making the density of the base sufficiently high.

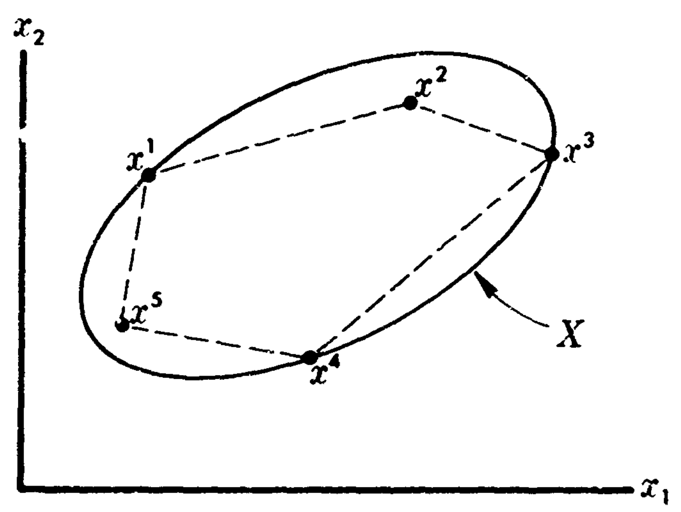

Fig. 2--Inner Linearization of a convex set

An example of Inner Linearization applied to a function of one variable is given in Fig. 3, where the function f has been approximated on the interval $[x^1,x^5]$ by a piecewise-linear function (represented by the dotted line) that accomplishes linear interpolation between the values of f at the base points $x^1,\ldots,x^5$. The approximation is "inner" in the sense that the epigraph of the approximating function lies entirely within the epigraph of the approximated function. (The *epigraph* of a convex (concave) function is the set of all points lying on or above (below) the graph of the function.)

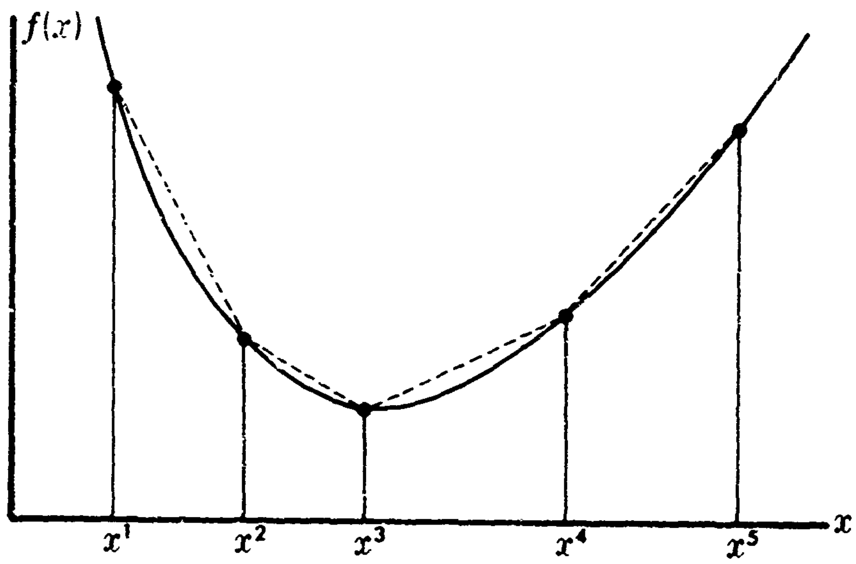

Fig. 3--Inner Linearization of a convex function

Let us further examine these two graphical examples of Inner Linearization in the context of the special problem

$$\text{(2.5)} \qquad \underset{x \in X}{\text{Minimize}} \ f(x) \ \text{ s.t. } \ G(x) \leq 0,$$

where n=2, X is a convex set, and all functions are convex. Inner-linearizing X as in Fig. 2 yields the approximation

$$\text{(2.6)}\quad \underset{\alpha \geqq 0}{\text{Minimize}} \quad f(\sum_{j=1}^{5} \alpha^j x^j) \quad \text{s.t.} \quad G(\sum_{j=1}^{5} \alpha^j x^j) \leqq 0, \ \sum_{j=1}^{5} \alpha^j = 1.$$

Note that the x variables are replaced by the "weighting" variables α^j, one for each chosen base point in X. Inner-linearizing f now as in the two-dimensional analog of Fig. 3 yields

$$\text{(2.7)}\quad \underset{\alpha \geqq 0}{\text{Minimize}} \quad \sum_{j=1}^{5} \alpha^j f(x^j) \quad \text{s.t.} \quad G(\sum_{j=1}^{5} \alpha^j x^j) \leqq 0, \ \sum_{j=1}^{5} \alpha^j = 1.$$

We have taken the bases for the approximations to X and f to coincide, since normally only one base is introduced for a given problem. An exception to this general rule may occur, however, when some of the functions are separable, for then it may be desirable to introduce different bases for different subsets of variables. Suppose, for example, that $f(x) = f_1(x_1) + f_2(x_2)$, $X = R^2$, and that we wish to use $< x_1^1,\dots,x_1^4 >$ as a base for inner-linearizing f_1 and $< x_2^1,\dots,x_2^6 >$ as a base for f_2. Then the corresponding approximation to (2.5) would be

$$\text{(2.8)}\quad \underset{\substack{\alpha_1 \geqq 0 \\ \alpha_2 \geqq 0}}{\text{Minimize}} \quad \sum_{j=1}^{4} \alpha_1^j f_1(x_1^j) + \sum_{j=1}^{6} \alpha_2^j f_2(x_2^j)$$

$$\text{s.t.} \quad G(\sum_{j=1}^{4} \alpha_1^j x_1^j, \ \sum_{j=1}^{6} \alpha_2^j x_2^j) \leqq 0, \ \sum_{j=1}^{4} \alpha_1^j = 1 \text{ and } \sum_{j=1}^{6} \alpha_2^j = 1.$$

Problems (2.6), (2.7) and (2.8) are all convex programs.

The general nature of Inner Linearization should be clear from these examples. It is important to appreciate that there is a great deal of flexibility in applying Inner Linearization--both as to which sets and functions are inner-linearized, and as to which base is used. Inner-linearizing everything results, of course, in a linear program, although it is by no means necessary to inner-linearize everything (see Sec. 4.3). The base can be chosen to approximate the set of points satisfying any subset whatever of the given constraints; the constraints in the selected subset are replaced by the simple non-negativity conditions on the weighting variables plus the normalization constraint, while the remaining constraints are candidates for functional Inner Linearization with respect to the chosen base. Or, if desired, the base can be chosen freely from the whole space of the decision variables (this can be thought of as corresponding to the selection of an empty set of constraints). Each of the given constraints, then, is placed into one of three categories, any of which may be empty: the constraints defining the convex set approximated by the chosen base, those that are inner-linearized over the base, and all others.

Inner Linearization has long been used for convex (or concave) functions of a single variable [Charnes and Lemke 54]. It has also been used for non-convex functions of a single variable [Miller 63]. Techniques based on this manipulation are sometimes called "separable programming" methods because they deal with functions that are linearly separable into functions of one variable (e.g., $f(x) \triangleq \sum_{i=1}^{n} f_i(x_i)$).

It is easy to determine--perhaps graphically--an explicit base yielding as accurate an inner-linearization as desired for a given function of one variable. It is much more difficult, however, to do

this for functions of many variables. Even if a satisfactory base could be determined, it would almost certainly contain a large number of points. This suggests the desirability of having a way to generate base points as actually needed in the course of computationally solving the inner-linearized problem. Hopefully it should be necessary to generate only a small portion of the entire base, with many of the generated points tending to cluster about the true optimal solution. Indeed there is a way to do this based on the solution strategy we call Restriction (Sec. 3.2). The net effect is that the Inner Linearization manipulation need only be done implicitly! Dantzig and Wolfe were the originators of this exceedingly clever approach to nonlinear programming [Dantzig 63a, Ch. 24]; we shall review this development in Sec. 4.3.

An important special case in which Inner Linearization can be used very elegantly concerns convex polytopes (the polytope could be the epigraph of a piecewise-linear convex function). Inner Linearization introduces no error at all in this case if the base is taken to coincide with the extreme points.† As above, the extreme points can be generated as needed if the implicitly inner-linearized problem is solved by Restriction. This is the idea behind the famous Decomposition Principle for linear programming [Dantzig and Wolfe 60], which is reviewed in Sec. 4.3.

For ease of reference in the sequel, the well-known theorem asserting the exactness of Inner Linearization for convex polytopes [Goldman 56] is recorded.

---

†It is also necessary, of course, to introduce the extreme rays if the polytope is unbounded.

Theorem 3. Any nonempty convex polytope $X \triangleq \{x : Ax \leqq b\}$ can be expressed as the vector sum $\mathcal{P} + C$ of a bounded convex polyhedron $\mathcal{P}$ and a cone $C \triangleq \{x: Ax \leqq 0\}$. $\mathcal{P}$ in turn can be expressed as the convex hull of its extreme vectors $< y_1, \ldots, y_p >$, and C can be expressed as the nonnegative linear combinations of a finite set of spanning vectors $< z_1, \ldots, z_q >$. (If $\mathcal{P}$ (respectively C) consists of only the 0-vector, take p (respectively q) equal to 0.) Thus there exist vectors $<y_1, \ldots, y_p; z_1, \ldots, z_q>$ such that $x \varepsilon X$ if and only if

$$x = \sum_{i=1}^{p} \alpha_i y_i + \sum_{i=1}^{q} \beta_i z_i$$

for some nonnegative scalars $\alpha_1, \ldots, \alpha_p$, $\beta_1, \ldots, \beta_q$ such that $\sum_{i=1}^{p} \alpha_i = 1$. Moreover, if the rank of A equals n (the number of its columns), then a representation with a minimal number of vectors is obtained by letting the y_i's be the extreme vectors of X and by letting the z_i's be distinct nonzero vectors in each of the extreme rays of C; this minimal representation is unique up to positive multiples of the z_i's.

It should be noted that in mathematical programming the rank of A usually equals n, since nonnegativity constraints on the variables are usually included in X. If this is not the case, then X can always be imbedded in the nonnegative orthant of R^{n+1} by a simple linear transformation (viz., put $x_i = y_i - y_o$, where $y_i \geq 0$, $i = 0, \ldots, n$).

There are also results having to do with economical inner linearizations of nonpolyhedral sets. For example, there is the Theorem of Krein and Milman [Berge 63, p. 167] that every closed, bounded, nonempty convex set is the convex hull of its extreme points. Usually, however, it suffices to know that a convex set or function can be represented as accurately as desired by Inner Linearization if a sufficiently dense base is chosen.

2.3 OUTER LINEARIZATION

Outer Linearization is complementary in nature to Inner Linearization, and also applies both to convex (or concave) functions and to convex sets.

An example as applied to a convex set in two dimensions is given by Fig. 4, where X has been approximated by a containing convex polytope that is the intersection of the containing half-spaces $H_1,\ldots,H_4$. The first three are actually supporting half-spaces that pass, respectively, through the points x^1, x^2, and x^3 on the boundary of X.

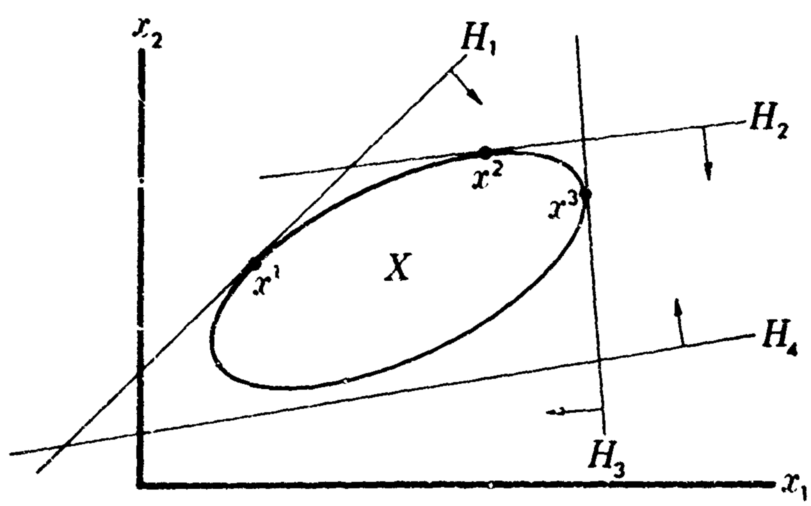

Fig. 4--Outer Linearization of a convex set

An example as applied to a function of one variable is given in Fig. 5, where the function f has been approximated by the piecewise-linear function that is the upper envelope, or pointwise maximum, of the linear supporting functions $s_1(x)$, ..., $s_5(x)$ associated with the points x^1, ..., x^5. A *linear support* for a convex function f at the point $\bar{x}$ is defined as a linear function with the property that it nowhere exceeds f in value, and equals f in value at $\bar{x}$.† The epigraph of the approximating function contains the epigraph of the approximated function when Outer Linearization is used.

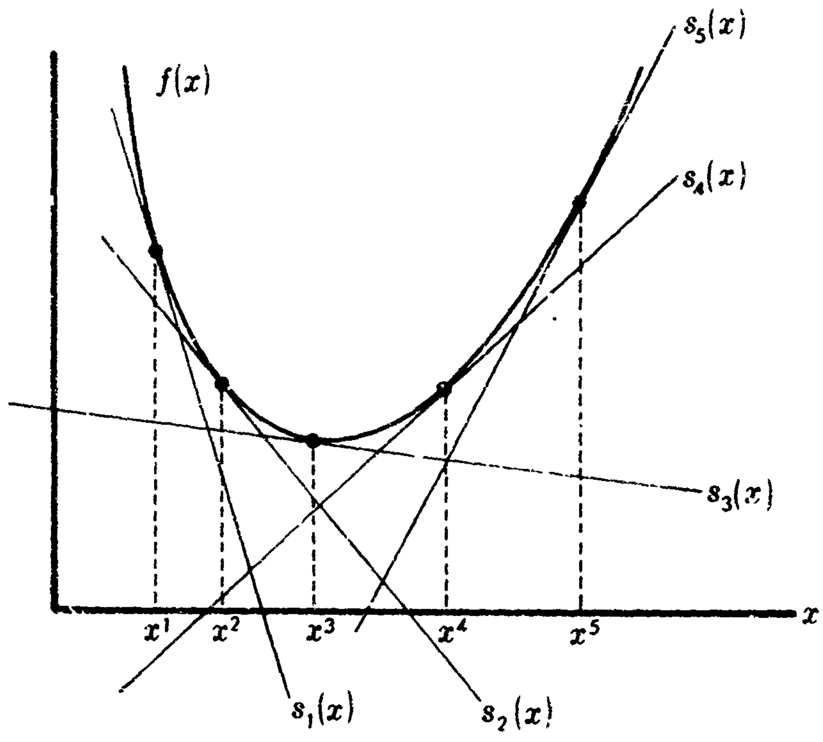

Fig. 5--Outer Linearization of a convex function

Obviously Outer Linearization is opposite to Inner Linearization in that it generally underestimates (overestimates) the value of a

---

†If f is differentiable at $\bar{x}$, then $f(\bar{x}) + \nabla f(\bar{x})(x - \bar{x})$ is a linear support at $\bar{x}$.

convex (concave) function, and includes not only the given convex set but points outside as well. The notion of conjugacy (see, e.g., [Rockafellar 68]) is a logical extension, but need not be pursued here.

That Outer Linearization truly linearizes a convex program like

$$\underset{x \in X}{\text{Minimize}}\ f(x) \quad \text{s.t.}\ G(x) \leqq 0 \tag{2.9}$$

should be clear. The approximation of X by a containing convex polytope can only introduce linear constraints; the approximation of g_i by the pointwise maximum of a collection of p_i linear supports, say, obviously leads to p_i linear inequalities; and the approximation of f by the pointwise maximum of p linear supports leads to p additional linear inequalities after one invokes the elementary manipulation of minimizing an upper bound on f in place of f itself.† If <u>all</u> nonlinear functions are dealt with in this fashion, the approximation to (2.9) is a linear program.

As with Inner Linearization, there is great latitude concerning which sets and functions are to be outer-linearized, and which approximants†† are to be used. In general, the objective function may or may not be outer-linearized, and each constraint is placed into one of three categories: the ones that together define a convex set to be outer-linearized, the ones that are outer-linearized individually, and the ones that are not outer-linearized at all.

---

†E.g., $\underset{x \in X}{\text{Min}}\ \text{Max}_i \{s_i(x)\} = \underset{\substack{x \in X \\ \sigma}}{\text{Min}}\ \sigma \quad \text{s.t.}\ \sigma \geq s_i(x),\ \text{all}\ i$.

††For the sake of unified terminology, we use the term *approximant* for a containing or supporting half-space of a convex set, and also for a linear bounding function or linear support of a convex function.

The main obstacle faced with Outer Linearization is that an excessive number of approximants may be required for an adequate approximation, especially for sets in more than two dimensions and functions of more than one variable. Fortunately, it turns out that it is usually possible to circumvent this difficulty, for there is a solution strategy applicable to the outer-linearized problem that enables approximants to be generated economically as needed without having to specify them in advance. We call this strategy Relaxation. The net effect is that the Outer Linearization manipulation need only be done implicitly. Two pioneering papers on this approach to nonlinear programming are [Kelley 60] and [Dantzig and Madansky 61]. Relaxation and the first of these papers are discussed in Sec. 3.3.

In large-scale programming, Outer Linearization is especially important in conjunction with Projection and Dualization. See, for example, the discussion of [Benders 62] in Sec. 4.1.

Approximation by Outer Linearization naturally raises the question of the existence of a supporting approximant at a given point. The main known result along these lines is that every boundary point of a convex set in R^n must have at least one supporting half-space passing through it. It follows that every closed convex set can be represented as the intersection of its supporting half-spaces [Berge 63, p. 166].[†] It also follows that every convex (or concave) function with a closed epigraph has a supporting half-space to its epigraph at every point where the function is finite. Unfortunately, this is not quite the same as the existence of a linear support at every such point, since the supporting

---

[†]Of course, a convex polytope by definition admits an exact outer-linearization using only a <u>finite</u> number of approximants.

half-space may be "vertical" when viewed as in Fig. 5. Various mild conditions could be imposed to preclude this kind of exceptional behavior, but for most purposes one may avoid the difficulty by simply working directly with the epigraph of a convex function.

3. SOLUTION STRATEGIES: SOURCE OF "SUBPROBLEMS"

The previous section described several prominent problem manipulations for restating a given problem in a more or less equivalent form. The result is often referred to in specific applications as a "master" problem. Typically one then applies a solution strategy designed to facilitate optimization by reduction to a sequence of simpler optimization problems. Quite often this leads to *subproblems* amenable to solution by specialized algorithms. There are perhaps a half dozen principal solution strategies, each applicable to a variety of problems and implementable in a variety of ways. This section presents three such strategies that seem to be especially useful for large-scale problems: the so-called Piecewise, Restriction and Relaxation strategies. See Table 2 for a classification of many known algorithms in terms of the solution strategy they can be viewed as using.

The Piecewise strategy is appropriate for problems that are significantly simpler if their variables are temporarily restricted to certain regions of their domain. The domain is (implicitly) subdivided into such regions, and the problem is solved by considering the regions one at a time. Usually it is necessary to consider only a small fraction of all possible regions explicitly. The development of the Piecewise strategy for large-scale programming is largely due to J. B. Rosen, whose various Partition Programming algorithms invoke it subsequent to the Projection manipulation.

Restriction is often appropriate for problems with a large number of nonnegative variables. It enables reduction to a recursive sequence of problems in which most of the variables are fixed at zero. The

Simplex Method itself turns out to be a special form of Restriction for linear programming, although the strategy also applies to nonlinear problems. Restriction is almost always used if Inner Linearization has been applied.

Relaxation is useful for problems with many inequality constraints. It reduces such a problem to a recursive sequence of problems in which many of these constraints are ignored. The Dual Method of linear programming is a special form of Relaxation, although the strategy applies equally well to nonlinear problems. Outer Linearization is almost always followed by Relaxation.

Perhaps the most important solution strategy <u>not</u> discussed here is the well-known <u>Feasible Direction</u> strategy [Zoutendijk 60], which reduces a problem with differentiable functions to a sequence of one-dimensional optimization problems along carefully chosen directions. Most of the more powerful primal nonlinear programming algorithms utilize this strategy, but their application to large-scale problems is frequently hampered by non-differentiability (if Dualization or Projection is used) if not by sheer size (especially if Inner or Outer Linearization is used). See Sec. 4.4 for an instance in which the first obstacle can be surmounted.

We have also omitted discussion of the <u>Penalty</u> strategy (e.g., [Fiacco and McCormick 68]), which reduces a constrained problem to a sequence of essentially unconstrained problems via penalty functions. The relevance of this strategy to large-scale programming is hampered by the fact that penalty functions tend to destroy linearity and linear separability.

3.1 PIECEWISE STRATEGY

Suppose that one must solve

$$\text{(3.1)} \qquad \underset{y \in Y}{\text{Maximize}}\ v(y),$$

where v is a "piecewise-simple" function (e.g., piecewise-linear or piecewise-quadratic) in the sense that there are regions (pieces) P^1, P^2, ... of its domain such that v coincides with a relatively tractable function v^k on P^k. The situation can be depicted as in Fig. 6, in which Y is a disk partitioned into four regions. Let us further suppose that v is concave on the convex set Y and that,

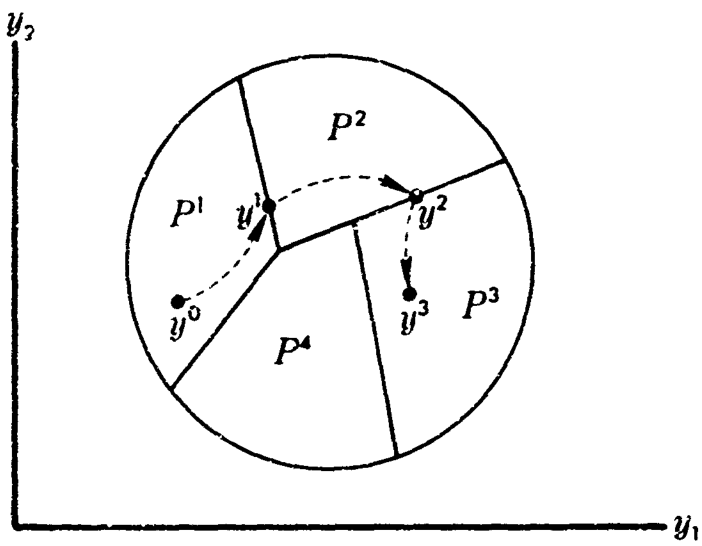

Fig. 6

given any particular point in Y, we can explicitly characterize the particular piece to which that point belongs, as well as v on that piece. Then it is natural to consider solving (3.1) in the following piecemeal fashion that takes advantage of the piecewise-simplicity of v. Note that it is unnecessary to explicitly characterize all of the pieces in advance.

<u>The Piecewise Strategy</u>

<u>Step 1</u> Let a point y^o feasible in (3.1) be given. Determine the corresponding piece P^o containing y^o and the corresponding function v^o.

<u>Step 2</u> Maximize $v^o(y)$ subject to $y \epsilon Y \cap P^o$. Let y' be an optimal solution (an infinite optimal value implies termination).

<u>Step 3</u> Determine a piece P' adjacent to P^o at $\bar{y}$ such that $v(y) > v(y^o)$ for some $y \epsilon Y \cap P'$ [if none exists, y^o is optimal in (3.1)]. Determine the corresponding function v' and return to Step 2 with P', v', and y' in place of P^o, v^o, and y^o.

A hypothetical trajectory for y is traced in Fig. 6 as a dotted line. Optimizations (Step 2) were performed in three regions before the optimal solution of (3.1) was found.

The problem at Step 2 has a simpler criterion function than (3.1) itself, although it has more constraints ($y \epsilon P^o$). If it is sufficiently simple by comparison with (3.1), then the Piecewise strategy is likely to be advantageous provided Steps 1 and 3 are not too difficult. Both Steps 2 and 3 can give rise to "subproblems" when this strategy is used for large-scale programming.

The principal use of the Piecewise strategy in large-scale programming is for problems resulting from Projection and Dualization. In both cases [cf. (2.2)], v involves the optimal value of an associated "inner" optimization problem parameterized by y. Evaluating v requires solving the inner problem, and so v is not explicitly available in closed form. Fortunately, it usually happens that evaluating $v(y^o)$ yields as a by-product a characterization of the piece P^o containing

y^o on which v has relatively simple form. We shall illustrate this with a simple example. See also Sec. 4.2 and [Geoffrion 68b; Sec. 5].

The Piecewise strategy can also be used to motivate a generalization of the Simplex Method that allows the minimand to be a sum of piecewise-linear univariate convex functions [Orden and Nalbandian 68].

Example

Constrained games and similar applications can lead to problems of the form

$$\text{(3.2)} \qquad \underset{y \varepsilon Y}{\text{Maximize}} \left[\underset{x \geq 0}{\text{Minimum}} \left\{ H^t(y)x \quad \text{s.t.} \quad Ax = b \right\} \right] ,$$

where $H(\cdot)$ is a concave vector-valued function on the convex set Y. The maximand of (3.2), v, is concave because it is the pointwise minimum of a collection of concave functions of **y**. Suppose that we evaluate v at $y^o \varepsilon Y$, with the corresponding optimal solution of the inner problem being x^o. The value is $H^t(y^o)x^o$. We know from the elementary theory of linear programming that, since changes in y cannot affect the feasibility of x^o, x^o remains an optimal solution of the inner problem as y varies so long as the "reduced costs" remain of the right sign. Hence the value of v(y) is $H^t(y)x^o$ for all y such that

$$\text{(3.3)} \qquad (H^B(y))^t B^{-1} A_{.j} - h_j(y) \leq 0, \text{ all nonbasic } j ,$$

where $A_{.j}$ is the j^{th} column of A, and the component functions of H^B correspond to the variables x_i in the optimal basis matrix B at y^o. Thus we see how to accomplish Step 1, and the problem to be solved at Step 2 is

$$\text{(3.4)} \qquad \underset{y \varepsilon Y}{\text{Maximize}}\ H^t(y)x^o \quad \text{s.t.} \quad (3.3)\ .$$

Note that (3.4) has the advantage over (3.2) of an explicit criterion function. Since $x^o \geq 0$, $H^t(\cdot)x^o$ is concave on Y.

Suppose that y' is an optimal solution of (3.4).[†] If y' is not optimal in (3.2), then there must be an alternate optimal basis B' at y' such that the corresponding problem (3.4) admits an improved solution. At worst, such an "improving" basis could be found by enumerating the alternative optimal bases at y'. At best, an improving basis would be revealed by a single active constraint among those of (3.3) at y'. One could also compute an improving feasible direction z' for (3.2) at y' (cf. Sec. 4.4); the appropriate improving basis would then be revealed by a parametric linear programming analysis of the inner problem.

3.2 RESTRICTION

Restriction is a solution strategy principally useful for problems with many nonnegative variables, the data associated with some of which perhaps being only implicitly available. Combinatorial models and Inner Linearization are two fertile sources of such problems.

The basic idea is as follows: solve the given problem subject to the additional restriction that a certain subset of the variables must have value 0; if the resulting solution does not satisfy the optimality

---

[†]It may be difficult to find a global optimum of (3.4) if H is not linear, for then (3.3) need not define a convex feasible region (unless $B^{-1}A_{.j} \leq 0$ for all nonbasic j). Fortunately, however, it can be seen from the concavity of v that a local optimum will generally suffice, although finite termination may now be in jeopardy.

conditions of the given problem, then "release" one or more restricted variables (allow them to be nonnegative) and so've this less-restricted problem; continue in this fashion until the optimality conditions of the given problem are satisfied, at which point the procedure terminates. An important refinement forming an integral part of the strategy involves adding variables to, as well as releasing them from, the restricted set. Note that the variables restricted to 0 essentially drop out of the problem, thereby reducing its size and avoiding the need for knowing the associated data explicitly. If (as is usually the case) only a fairly small proportion of all variables actually are active (positive) at an optimal solution, then this strategy becomes quite attractive.

The earliest and most significant embodiment of the Restriction strategy turns out to be the Simplex Method for linear programming itself. It can be shown, as we shall indicate, that a natural specialization of Restriction to the completely linear case yields the very same sequence of trial solutions as does the ordinary Simplex Method. All of the column-generation schemes for implementing the Simplex Method for linear programs with a vast number of variables can therefore be viewed in terms of Restriction. We shall review one of these schemes [Gilmore and Gomory 61] at the end of this section.† The usefulness of Restriction is not, however, limited to the domain of linear programming. It will be shown in Sec. 4.3 how this strategy can yield, in a nonlinear case, variations of the Dantzig-Wolfe method for convex programming.

---

†Another column-generating scheme is explained in Sec. 4.3. See also part B of Table 1.

Formal Statement

Consider the problem

$$\text{(3.5)} \qquad \underset{x \in X}{\text{Maximize}}\ f(x) \text{ s.t. } g_i(x) \geq 0,\ i = 1, \ldots, m,$$

where f is a concave function on the nonempty convex set $X \subseteq R^n$ and the functions $g_1, \ldots, g_m$ are all linear. All nonlinear constraints, as well as any linear constraints that are not to be restricted, are presumed to be incorporated in X. The typical restricted version of (3.5) is the (still concave) problem

$$\text{(3.6)} \qquad \underset{x \in X}{\text{Maximize}}\ f(x) \text{ s.t. } \begin{array}{l} g_i(x) = 0,\ i \in S \\ g_i(x) \geq 0,\ i \notin S, \end{array}$$

where S is a subset of the m constraint indices. [Note that we are presenting Restriction in a seemingly more general setting than the motivational one above in that general linear inequality constraints, as well as simple variable nonnegativities, are allowed to be restricted to equality. Actually, the present setting is no more general since slack variables could be introduced to accommodate the restriction of general linear inequalities.] Some, none, or all of the $x_j \geq 0$ type constraints (if any) may be included among $g_1, \ldots, g_m$. The analyst is free to choose the linear inequality constraints to associate with X; the rest are candidates for restriction.

An optimal solution of the restricted problem (3.6) will be denoted by x^S, and a corresponding optimal multiplier vector (which, under mild assumptions, must exist) is denoted by $\mu^S = (\mu_1^S, \ldots, \mu_m^S)$. The pair (x^S, μ^S) satisfies the Kuhn-Tucker optimality conditions for (3.6), namely

(i) x^S maximizes $f(x) + \sum_{i=1}^{m} u_i^S g_i(x)$ over X

(ii) x^S is feasible in (3.6)

(iii) $u_i^S \geq 0$, $i \notin S$

(iv) $u_i^S g_i(x^S) = 0$, $i \notin S$.

We are now ready to give a formal statement of Restriction applied to (3.5). Notice that not only are constraints released from the current restricted set S at each iteration, but additions are also made whenever $g_i(x^S) = 0$ for some $i \notin S$, provided that $f(x^S)$ has just increased.

The Restriction Strategy

Step 1 Put $\bar{f} = -\infty$ and S equal to any subset of indices such that the corresponding restricted problem (3.6) is feasible.

Step 2 Solve (3.6) for an optimal solution x^S and associated optimal multipliers u_i^S (if it has unbounded optimal value, the same must be true of the given problem (3.5) and we terminate). If $u_i^S \geq 0$ for all $i \in S$, then terminate (x^S is optimal in (3.5)); otherwise, go on to Step 3.

Step 3 Put V equal to any subset of S that includes at least one constraint for which $u_i^S < 0$. If $f(x^S) > \bar{f}$, replace $\bar{f}$ by $f(x^S)$ and S by E-V, where $E \triangleq \{1 \leq i \leq m : g_i(x^S) = 0\}$; otherwise, (i.e., if $f(x^S) = \bar{f}$), replace S by S-V. Return to Step 2.

We assume that the given problem (3.5) admits a feasible solution, so that Step 1 is possible. To ensure that Step 2 is always possible, we also assume that the restricted problem (3.6) admits an optimal solution and multiplier vector whenever it is feasible and has finite supremal value. It is a straightforward matter to show that the termination conditions of Step 2 are valid, and Step 3 is obviously

always possible. Thus the strategy is well defined, although we have deliberately not specified how to carry out each step.

An important property is that the sequence $<f(x^S)>$ is non-decreasing. Thus the strategy yields an improving sequence of feasible solutions to (3.5). Moreover, $<f(x^S)>$ can be stationary in value at most a finite number of consecutive times, since the role of $\bar{f}$ at Step 3 is to insure that S is augmented (before deletion by V) only when $f(x^S)$ has just increased. Hence termination must occur in a finite number of steps, for there is only a finite number of possibilities for S and each increase in $f(x^S)$ precludes repetition of any previous S.

Options and Relation to the Simplex Method

Let us now consider the main options of Restriction beyond the decision as to which of the linear inequality constraints will comprise $g_1, \ldots, g_m$.

(i) How to select the initial S at Step 1?

(ii) How to solve (3.6) for (x^S, u^S) at Step 2?

(iii) What criterion to use in selecting V at Step 3?

How these options are exercised exerts a great influence upon the efficiency.

As stated above, there is an intimate relationship between Restriction and the Simplex Method in the completely linear case. Given the linear program

$$\underset{x}{\text{Maximize}}\ c^t x \quad \text{s.t.} \quad Ax = b,\ x \geq 0,$$

define (3.5) according to the identifications

$$f(x) = c^t x$$

$$g_i(x) = x_i, \text{ all } i$$

$$X = \{x : Ax = b\},$$

and specialize Restriction as follows: let the initial S be chosen to coincide with the nonbasic variables in an initial basic feasible solution, and select V at Step 3 to be the index of the most negative u_i^s. It can then be shown, under the assumption of nondegeneracy, that Restriction is equivalent to the usual Simplex Method in that the set of nonbasic variables at the ν^{th} iteration of the Simplex Method necessarily coincides with $\bar{S}$ at the ν^{th} iteration of Restriction, and the ν^{th} basic feasible solution coincides with the ν^{th} optimal solution x^s of (3.6). Thus Restriction can be viewed as one possible strategic generalization of the Simplex Method. Not only is this an interesting fact in its own right, but it also permits us to draw some inferences--as we shall see in the discussion below--concerning how best to exercise the options of Restriction.

<u>Step 1</u>

The selection of the initial S should be guided by two objectives: to make the corresponding restricted problem easy to solve by comparison with the given problem, and to utilize any prior knowledge that may be available concerning which of the g_i constraints are likely to hold

with equality at an optimal solution. In the Simplex Method, for example, the initial choice of S implies that the restricted problem is trivial since it has a unique feasible solution; at every subsequent execution of Step 2, the restricted problem remains nearly trivial with essentially only one free variable (the entering basic variable). Useful prior knowledge is often available if the given problem is amenable to physical or mathematical insight or if a variant has been solved previously.

Step 2

How to solve the restricted problem for (x^s, u^s) at Step 2 depends, of course, on its structure. Hopefully, enough constraints will be restricted to equality to make it vastly simpler than the original problem. In any event, it is advisable to take advantage of the fact that a sequence of restricted problems must be solved as the Restriction strategy is carried out. Except for the first execution of Step 2, then, what is required is a solution recovery technique that effectively utilizes the previous solution. The pivot operation performs precisely this function in the Simplex Method, and serves as an ideal to be approached in nonlinear applications of Restriction.

It is worth mentioning that many solution (or solution recovery) techniques that could be used for the restricted problem automatically yield μ^s as well as x^s. When this is not the case, one may find μ^s once x^s is known by solving a linear problem if f and the constraint functions defining X are differentiable, since under these conditions the Kuhn-Tucker optimality conditions for (3.6) in differential form become linear in μ.

Step 3

Perhaps the most conspicuous criterion for choosing V at Step 3 is to let it be the index of the constraint corresponding to the most negative μ_i^s. One rationale for this criterion is as follows. Suppose that μ^s is unique. It can then be shown (see [Geoffrion 69] or [Rockafellar 68]) that the optimal value of the restricted problem is differentiable as a function of perturbations about 0 of the right-hand side of the g_i constraints, and that $-\mu_i^s$ is the partial derivative of the optimal value with respect to such perturbations of the i^{th} constraint. Thus the most negative μ_i^s identifies the constraint in S whose release will lead to the greatest initial rate of improvement in the value of f as this constraint is permitted to deviate positively from strict equality. It can be argued that μ^s is likely to be unique, but if we drop this supposition then $-\mu_i^s$ still provides an upper bound on the initial rate of improvement even though differentiability no longer holds.

This most-negative-multiplier criterion is precisely the usual criterion used by the Simplex Method in its version of Step 3 to select the entering basic variable, but it is by no means the only criterion used. The extensive computational experience presently available with different criteria used in the Simplex Method may permit some inferences to be drawn concerning the use of analogous criteria in the nonlinear case. It has been observed [Wolfe and Cutler 63], for example, that the most-negative-multiplier criterion typically leads to a number of iterations equal to about twice the number of constraints, and that other plausible criteria can be

expected to be consistently better by no more than a factor of two or so.[†] Lest it be thought that V must necessarily be a singleton, we note that we may interpret Wolfe and Cutler to have also observed [ibid., p. 190] that choosing V to consist of the five most negative multipliers reduced the number of iterations by a factor of two as compared with the single-most-negative-multiplier choice.[††] Of course, this increases the time required to solve each restricted problem. Experience such as this should at least be a source of hypotheses to be examined in nonlinear applications of Restriction.

Mechanizing the "Pricing" Operation

Each iteration of Restriction requires determining whether there exists a negative multiplier and, if so, at least one must be found. In the ordinary Simplex Method, which as has been indicated can be viewed as a particular instance of Restriction, this was originally done enumeratively by scanning the row of reduced costs for an entry of the "wrong" sign. To deal with large numbers of variables, however, it is desirable whenever possible to replace this enumeration by an algorithm that exploits the structure of the problem. This is referred to as *mechanized pricing*.

[†] An example of another plausible criterion is this: select V to be the index of the constraint which, when deleted from S, will result in the greatest possible improvement in the optimal value of the restricted problem. Of course, this criterion is likely to be prohibitively expensive computationally in the nonlinear case.

[††] This is known as *multiple pricing*, a feature used in most production linear programming systems designed for large-scale problems. See, for example, [Orchard-Hays 68, Sec 6.1].

Mechanized pricing is widely practiced in the context of linear programming, where it is often referred to as *column-generation*. Since the pioneering paper [Ford and Fulkerson 58], many authors have shown how pricing could be mechanized by means of subsidiary network flow algorithms, dynamic programming, integer programming, and even linear programming. See the references of part B of Table 1, [Balinski 64], and [Gomory 63]. It will suffice to mention here but one specific illustration: the cutting-stock problem as treated by [Gilmore and Gomory 61]. See also Sec. 4.3.

Cutting-Stock Problem

A simple version of Gilmore and Gomory's cutting-stock problem, without the integrality requirement on x, is

$$\text{(3.7)} \qquad \underset{x \geq 0}{\text{Minimize}} \sum_j x_j \quad \text{s.t.} \quad \sum_j a_{ij} x_j \geq r_i, \; i = 1, \ldots, m,$$

where a_{ij} is the number of pieces of length ℓ_i produced when the cutting knives are set in the j^{th} pattern, r_i is the minimum number of required pieces of length ℓ_i, and x_j is the number of times a bar of stock is cut according to pattern j. The number of variables is very large because of the great variety of ways in which a bar of stock can be cut. It is easy to see that each column of the matrix A is of the form $(y_1, \ldots, y_m)^t$, where y is a vector of nonnegative integers satisfying $\sum_{i=1}^{m} \ell_i y_i \leq \lambda$ (λ is the length of a bar of

stock); and conversely, every such vector corresponds to some column (assuming that all possible patterns are allowed). When Restriction is applied to (3.7) in the form of the Simplex Method, it follows that the problem of determining the most negative multiplier can be expressed as the subsidiary optimization problem

$$(3.8) \qquad \underset{y \geq 0}{\text{Minimize}}\ 1 - u^t y \quad \text{s.t.} \sum_{i=1}^{n} \ell_i y_i \leq \lambda,\ y \text{ integer},$$

where u is the known vector of the current "Simplex multipliers." If slack variables are given priority over structural variables in determining entering basic variables (cf. Sec. 4.3), then u can be assumed nonnegative and (3.8) is a problem of the well-known "knapsack" variety, for which very efficient solution techniques are available. See [Gilmore and Gomory 61] for full details.

3.3 RELAXATION

Whereas Restriction is a solution strategy principally useful for problems with a large number of variables, the complementary strategy of *Relaxation* is primarily useful for problems with a large number of inequality constraints, some of which may be only implicitly available. Such problems occur, for example, as a result of Outer Linearization.† One of the earliest uses of Relaxation was in [Dantzig, Fulkerson, and Johnson 54], and since that time this strategy has appeared in one guise

†Relaxation can also be useful for dealing with large numbers of nonnegative variables; when a constraint such as $x_j \geq 0$ is relaxed, the variable x_j can often be substituted out of the problem entirely [Ritter 67c], [Webber and White 68].

or another in the works of numerous authors.[†] We discuss [Kelley 60] at the end of this section, and [Benders 62] in Sec. 4.1.

The essential idea of Relaxation is this: solve a relaxed version of the given problem ignoring some of the inequality constraints; if the resulting solution does not satisfy all the ignored constraints, then generate and include one or more violated constraints in the relaxed problem and solve it again; continue in this fashion until a relaxed problem solution satisfies all of the ignored constraints, at which point an optimal s ' of the given problem has been found. An important refinement invo_ 'ropping ımply satisfied constraints from the relaxed problem when this does not destroy the inherent finiteness of the procedure. We give a formal statement of Relaxation (with the refinement) below.

Relaxation and Restriction are complementary strategies in a very strong sense of the word. In linear programming, for example, whereas a natural specialization of Restriction is equivalent to the ordinary Simplex Method, it is also true [Geoffrion 68a] that a similar specialization of Relaxation is equivalent to Lemke's Dual Method. It follows, very significantly, that Restriction (Relaxation) applied to a linear program essentially corresponds to Relaxation (Restriction) applied to the dual linear program. In fact [ibid.], the same assertion holds for quite general convex programs as well. This complementarity makes it possible to translate most statements about Restriction into statements about Relaxation, and conversely.

[†]Re axation without problem manipulation is used in Dantzig 55a, Sec. 3; Stone 58; Thompson, Tonge and Zionts 66; Ritter 67c; Grigoriadis and Ritter 68. The following papers all use the pattern Outer Linearization/Relaxation: Cheney and Goldstein 59; Kelley 60; Dantzig and Madansky 61, p. 174; Parikh 67; Veinott 67. The references of part A of Table 2 all use the pattern Projection, Outer Linearization/Relaxation. See also the second footnote in Sec. 1.2.

Since we have already given a relatively detailed discussion of Restriction, a somewhat abbreviated discussion of Relaxation will suffice. See [ibid.] for a more complete discussion.

Formal Statement

Let f, g_1, ..., g_m be concave functions on a nonempty convex set $X \subseteq R^n$. The concave program

$$\text{(3.9)} \qquad \text{Maximize}_{x \in X} \; f(x) \text{ s.t. } g_i(x) \geq 0,\ i = 1, \ldots, m$$

is solved by solving a sequence of relaxed problems of the form

$$\text{(3.10)} \qquad \text{Maximize}_{x \in X} \; f(x) \text{ s.t. } g_i(x) \geq 0,\ i \in S,$$

where S is a subset of $\{1, \ldots, m\}$. Assume that (3.10) admits an optimal solution x^S whenever it admits a feasible solution and its maximand is bounded above on the feasible region, and assume further that an initial subset of constraint indices is known such that (3.10) has a finite optimal solution. (This assumption can be enforced, if necessary, by enforcing continuity of all functions and compactness of X.)

Under these assumptions, it is not difficult to show that the following strategy is well defined and terminates in a finite number of steps with either an optimal solution of the given problem (3.9) or knowledge that none exists; moreover, in the first case a nonincreasing sequence $\langle f(x^S)\rangle$ of upper bounds on the optimal value of (3.9) is obtained and the first solution of (3.10) that is feasible in (3.9) is also optimal. This version of Relaxation deletes amply satisfied constraints from S so long as $\langle f(x^S)\rangle$ is decreasing.

The Relaxation Strategy

<u>Step 1</u> Put $\overline{f} = \infty$ and S equal to any subset of indices such that the corresponding relaxed problem (3.10) has a finite optimal solution.

<u>Step 2</u> Solve (3.10) for an optimal solution x^S if one exists; if none exists (i.e., if the relaxed problem is infeasible), then terminate (the given problem is infeasible). If $g_i(x^S) \geq 0$ for all $i \notin S$, then terminate (x^S is optimal in the given problem); otherwise, go on to Step 3.

<u>Step 3</u> Put V equal to any subset of constraint indices that includes at least one constraint such that $g_i(x^S) < 0$. If $f(x^S) < \overline{f}$, replace $\overline{f}$ by $f(x^S)$ and S by $E \cup V$, where $E \triangleq \{i \in S : g_i(x^S) = 0\}$; otherwise (i.e., if $f(x^S) = \overline{f}$), replace S by $S \cup V$. Return to Step 2.

<u>Discussion</u>

As with Restriction, the analyst has considerable leeway concerning how he applies the Relaxation strategy. For instance, he can select the constraints that are to be candidates for Relaxation ($g_1, \ldots, g_m$) in any way he wishes; the rest comprise X. He is free to choose the initial S so as to allow an easy start, or to take advantage of prior knowledge concerning which of the constraints might be active at an optimal solution. He can choose the most effective solution recovery technique to reoptimize the successive relaxed problems. And, very importantly, he can choose the criterion by which V will be selected at Step 3 and the method by which the criterion will be implemented.

Probably the most natural criterion is to let V be the index of the most violated constraint. This is the criterion most commonly employed in the Dual Method of linear programming, for example,

although other criteria are possible. The complementarity between Relaxation and Restriction mentioned earlier enables us to interpret existing computational experience in linear programming so as to shed light on the merits and demerits of several alternative criteria. The discussion of Step 3 of Restriction should make further discussion of this point unnecessary. We should remark, however, that in some applications (e.g., [Dantzig, Fulkerson and Johnson 54], [Gomory 58], [Kelley 60]) only one or a few violated constraints are accessible each time the relaxed problem is solved, and it is therefore indicated that these be used regardless of whether they satisfy any particular criterion. In other applications a criterion such as "most violated constraint" is within the realm of attainability, and can be approached via a subsidiary linear program [Benders 62], network flow problem [Gomory and Hu 62], or some other subsidiary optimization problem that is amenable to efficient solution. This is the counterpart of mechanized pricing in Restriction.

Restriction and Relaxation, opposites though they are to one another, are by no means incompatible. In fact it can be shown [Geoffrion 66 and 67] that both strategies can be used simultaneously. The reduced problems become still more manageable, but assurance of finite termination requires more intricate control.

The Cutting-Plane Method

One important use of Relaxation occurs, as we have mentioned, in connection with problems that have been outer-linearized. This

will be illustrated in the simplest possible setting in terms of the problem

$$\text{(3.11)} \qquad \underset{x \geq 0}{\text{Minimize}}\ c^t x \ \text{ s.t. } \ Ax \leq b, \quad g(x) \leq 0,$$

where g is a convex function that is finite-valued on

$$X \triangleq \{x \geq 0 : Ax \leq b\}.$$

If one manipulates (3.11) by invoking an arbitrarily fine outer-linearization of g and then applies the Relaxation strategy with the new approximating constraints as the candidates for being relaxed, the resulting procedure is that of [Kelley 60].

Let us assume for simplicity that g is differentiable on X.[†] Then g has a linear support $g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x})$ at every point $\bar{x}$ in X, where $\nabla g(\bar{x})$ is the gradient of g at $\bar{x}$, and so (3.11) is equivalent to

$$\text{(3.12)} \qquad \underset{x \in X}{\text{Minimize}}\ c^t x \ \text{ s.t. } \ g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0, \text{ all } \bar{x} \in X.$$

The Relaxation strategy is the natural one for solving (3.12), since it avoids the need to determine in advance all of the linear supports of g. At each iteration, a relaxed version of this problem with a finite number of approximating constraints is solved. The optimal solution $\hat{x}$ of the relaxed problem is feasible in (3.12) if

[†]The assumption of differentiability can be weakened, since it is only necessary for g to have a support at each point of X. And even this requirement can be weakened as implicitly suggested in the conclusion of Sec. 2.3 if (3.11) is phrased in terms of the epigraph of g.

and only if $g(\hat{x}) \leq 0$; if $g(\hat{x}) > 0$, then evaluation of $\nabla g(\hat{x})$ yields a violated constraint that must be appended to the current relaxed problem. Since each relaxed problem is a linear program that will be augmented by a violated constraint, it is natural to reoptimize it using postoptimality techniques based on the Dual Method for linear programming.

It is easy to generalize this development to cover the case in which (3.11) has several (nonlinear) convex constraints and a convex minimand.

It should be pointed out that dropping amply satisfied constraints from the relaxed problem--a feature incorporated in our statement of Relaxation--is questionable in this context since (3.12) has an infinite number of constraints. Without this feature, Kelley has given mild conditions under which convergence to an optimal solution of (3.11) is assured in the limit.

We remark in passing that the approach of [Hartley and Hocking 63] for (3.11) can be viewed as Restriction applied to the dual of (3.12). Since Relaxation of (3.12) corresponds to Restriction of its dual, the two approaches are really equivalent.

4. SYNTHESIZING ALGORITHMS FROM MANIPULATIONS AND STRATEGIES

This section further illustrates the problem manipulations and solution strategies of the previous two sections by combining them in various ways to yield several known algorithms. The main object is not an exposition of these algorithms, although this is certainly important; rather, we wish to focus on the principal *patterns* in which manipulations and strategies can be assembled. These patterns constitute the real common denominators in the literature on large-scale programming. See Table 2.

It is beyond the scope of this effort to exemplify all of the important patterns of manipulations and strategies. We shall limit our discussion to five key ones:

1. PROJECTION, OUTER LINEARIZATION/RELAXATION
2. PROJECTION/PIECEWISE
3. INNER LINEARIZATION/RESTRICTION
4. PROJECTION/FEASIBLE DIRECTIONS
5. DUALIZATION/FEASIBLE DIRECTIONS

The first pattern is illustrated in Sec. 4.1 by Benders' Partitioning Procedure for what might be called semilinear programs; the second is illustrated in Sec. 4.2 by Rosen's Primal Partition Programming algorithm for linear programs with block-diagonal structure; the third in Sec. 4.3 by Dantzig-Wolfe Decomposition; the fourth in Sec. 4.4 by a procedure the author recently developed for nonlinear programs with multidivisional structure; and the fifth in Sec. 4.5 by the "local" approach discussed by Takahashi for concave programs with "complicating"

constraints. Another key pattern, OUTER LINEARIZATION/RELAXATION, was already illustrated in Sec. 3.3 with reference to Kelley's cutting-plane method. In addition, it is indicated in Sec. 4.2 how Rosen's algorithm can be used to illustrate the pattern DUALIZATION/PIECEWISE, and in Sec. 4.3 how Dantzig-Wolfe Decomposition can be used to illustrate DUALIZATION, OUTER LINEARIZATION/RELAXATION.

The discussion of the various algorithms is as uncluttered by detail as we have been able to make it. There is little or no mention of how to find an initial feasible solution,† the details of computational organization, or questions of theoretical convergence. The reader is invited to ponder such questions in the light of the concepts and results advanced in the previous two sections, and then to consult the original papers.

4.1 [Benders 62]

One might refer to

$$\text{(4.1)} \qquad \underset{\substack{x \geq 0 \\ y \in Y}}{\text{Maximize}}\; c^t x + f(y) \text{ s.t. } Ax + F(y) \leq b$$

as a *semilinear* program because it is a linear program in x when y is held fixed temporarily. The algorithm of [Benders 62] for this problem can be recovered by applying the pattern PROJECTION, OUTER LINEARIZATION/RELAXATION. Specifically, project (4.1) onto the space of the y variables, outer-linearize the resulting supremal value function in the maximand, and apply the Relaxation strategy

†If one exists, it can usually be found by applying the algorithm itself to a suitably modified version of the given problem.

to the new constraints arising as a consequence of Outer Linearization. Assume for simplicity that (4.1) is feasible and has finite optimal value.

Projection onto the space of the y variables yields

$$(4.2) \qquad \underset{y \in Y}{\text{Maximize}} \left[f(y) + \underset{x \geqq 0}{\text{Sup}} \{c^t x \text{ s.t. } Ax \leqq b - F(y)\}\right].$$

Note that the supremal value function appearing in the maximand corresponds to the linear program

$$(4.3) \qquad \underset{x \geqq 0}{\text{Maximize}}\ c^t x \text{ s.t. } Ax \leqq b - F(y).$$

This program is parameterized nonlinearly in the right-hand side by y, and our assumption implies that it has a finite optimum for at least one value of y. By the Dual Theorem, therefore, the dual linear program

$$(4.4) \qquad \underset{u \geqq 0}{\text{Minimize}}\ u^t(b - F(y)) \text{ s.t. } u^t A \geqq c^t$$

must be feasible (for all y). Let $\langle u^1, \ldots, u^p\rangle$ be the extreme points and $\langle u^{p+1}, \ldots, u^{p+q}\rangle$ representatives of the extreme rays of the feasible region of (4.4) (cf. Th. 3). Again using the Dual Theorem, we see that (4.3) is feasible if and only if (4.4) has finite optimal value, that is, if and only if y satisfies the constraints

$$(4.5) \qquad (u^j)^t(b - F(y)) \geq 0, \; j = p + 1, \ldots, p + q.$$

Since we take the supremal value function in (4.2) to be $-\infty$ for y such that (4.3) is infeasible--see Sec. 2.1--we may append the

constraints (4.5) to (4.2). Thus Projection applied to (4.1) yields (4.2) subject to the additional constraints (4.5).

Next we outer-linearize the supremal value function appearing in (4.2). It is easy to see, referring to (4.4), that its value is precisely

$$\text{(4.6)} \qquad \underset{1 \le j \le p}{\text{Minimum}} \ \{(u^j)^t(b - F(y))\}$$

for all y feasible in (4.2) with (4.5) appended. (Strictly speaking, it is accurate to call this Outer "Linearization" only if F is linear.) With this manipulation, (4.2) becomes

$$\text{(4.7)} \qquad \underset{y \in Y}{\text{Maximize}} \ [f(y) + \underset{1 \le j \le p}{\text{Minimum}} \ \{(u^j)^t(b - F(y))\}] \text{ s.t. (4.5)}$$

or, with the help of an elementary manipulation based on the fact that a minimum is really a greatest lower bound,

$$\text{(4.8)} \quad \underset{\substack{y \in Y \\ y_o}}{\text{Maximize}} \ f(y) + y_o$$

$$\text{s.t. } y_o \le (u^j)^t(b - F(y)), \ j = 1, \ldots, p$$

$$(u^j)^t(b - F(y)) \ge 0, \ j = p + 1, \ldots, p + q.$$

This is the master problem to be solved.

Relaxation is a natural strategy for (4.8); it avoids having to determine in advance all of the vectors u^j, $j = 1, \ldots, p + q$. To test the feasibility of a trial solution $(\hat{y}_o, \hat{y})$, where $\hat{y} \in Y$, one solves the linear subproblem (4.4) with y equal to $\hat{y}$. If the infimal value is greater than or equal to $\hat{y}_o$, then $(\hat{y}_o, \hat{y})$ is feasible and therefore

optimal in (4.8); $\hat{y}$, along with $\hat{x}$ equal to the optimal dual variables of (4.4), is an optimal solution of the given problem (4.1). If, on the other hand, the infimal value is less than $\hat{y}_o$, then a violated constraint of (4.8) is produced (some u^j with $1 \leq j \leq p$ is found if the infimal value is finite, while $p + 1 \leq j \leq p + q$ if it is $-\infty$). Of course, f, F, and Y must satisfy the obvious convexity assumptions if dropping amply satisfied constraints is to be justified. These assumptions will probably have to hold anyway if the relaxed problems based on (4.8) are to be concave programs (remember $u^j \geq 0$). There is, however, at least one other interesting case: if Y is a discrete set, say the integer points of some convex polytope, while f and F are linear, then (4.8) is a pure (except for y_o) integer linear program (see [Balinski and Wolfe 63], [Buzby, Stone and Taylor 65]).

The present development seems preferable to the original one since: (a) it justifies dropping amply satisfied constraints from successive relaxed versions of (4.8); (b) it retains f(y) in its natural position in the criterion function of (4.8) (Benders' version of (4.8), which is also equivalent to (4.7), has y_o alone as the criterion function and an added term f(y) in the right-hand side of each of the first p constraints); and (c) its comparative simplicity suggests a generalization, with the help of nonlinear duality theory, permitting nonlinearities in x. Details concerning (c) will be provided in a forthcoming paper.

4.2 [Rosen 64]

The algorithm of [Rosen 64] for the linear program

$$(4.9)\quad \underset{x,y}{\text{Maximize}}\ b_o^t y + \sum_{i=1}^{\ell} b_i^t x_i \text{ s.t. } x_i^t A_i + y^t D_i \leqq c_i^t,\ i = 1, \ldots, \ell$$

illustrates the pattern PROJECTION/PIECEWISE. Assume for simplicity that (4.9) is feasible and has finite optimal value.

Projection onto the y variables yields the master problem

$$(4.10)\quad \underset{y}{\text{Maximize}} \left[b_o^t y + \sum_{i=1}^{\ell} \underset{x_i}{\text{Sup}} \left\{ b_i^t x_i \text{ s.t. } x_i^t A_i \leqq c_i^t - y^t D_i \right\} \right],$$

where we have separated the supremum in the maximand (this separation is perhaps the main justification for using Projection).

The Piecewise strategy is appropriate for (4.10) because each supremal value in the maximand is piecewise-linear as a function of y. This follows from the elementary theory of linear programming, as we now explain. Let $\hat{y}$ be feasible in (4.10) in the sense that the maximand is not $-\infty$. Then each of the ℓ linear programs appearing in the maximand must have a finite optimal value, and by the Dual Theorem this optimal value must be equal to that of the dual linear program

$$(4.11)\quad \underset{u_i \geqq 0}{\text{Minimize}}\ (c_i^t - \hat{y}^t D_i) u_i \text{ s.t. } A_i u_i = b_i.$$

Let the vector $\hat{u}_i$ be an optimal solution of this program, and let the corresponding basis matrix be B_i. Since changes in y cannot affect the feasibility of $\hat{u}_i$, the optimal value of (4.11)--which is equal to the value of i^{th} supremal value function of (4.10) at y--must be

$$(c_i^t - y^t D_i)\hat{u}_i \tag{4.12}$$

so long as the reduced costs remain of the correct sign, that is, so long as y satisfies the condition

$$(c_i^t - y^t D_i)^{B_i} B_i^{-1}(A_i)_{\cdot j} - (c_i^t - y^t D_i)_j \le 0, \text{ all nonbasic } j, \tag{4.13}$$

where the superscript B_i masks all but the basic components of $(c_i^t - y^t D_i)$. Thus the master problem (4.10), confined to the linear "piece" containing $\hat{y}$, becomes the linear program

$$\underset{y}{\text{Maximize}}\ b_o^t y + \sum_{i=1}^{\ell} (c_i^t - y^t D_i)\hat{u}_i \text{ s.t. (4.13), } i = 1, \ldots, \ell. \tag{4.14}$$

This shows that Step 2 of the Piecewise strategy can be accomplished by linear programming. Rosen actually works with the dual of (4.14). His Theorems 1 and 2 concern Step 3 (cf. the discussion following (3.4) in Sec. 3.1).

It is interesting to note that if we had started with the dual of (4.9)—a block-diagonal linear program with coupling constraints—we would have obtained precisely the same procedure as the one just described by dualizing with respect to the coupling constraints only [Geoffrion 69] and then invoking the Piecewise strategy. In this way [Rosen 64] could also be used to illustrate the pattern DUALIZATION/PIECEWISE.

4.3 DANTZIG-WOLFE DECOMPOSITION

Dantzig-Wolfe Decomposition is archetypical of the pattern INNER LINEARIZATION/RESTRICTION. Mechanized pricing plays a prominent role. We shall illustrate this pattern first with the algorithm of [Dantzig and Wolfe 60] for a purely linear program, then with the algorithm of [Dantzig 63a, Ch. 24] for a nonlinear program, and finally with a variation of the latter in which not all nonlinear functions need be inner-linearized.

It is interesting to note that Dantzig-Wolfe Decomposition can also be viewed as an instance of the pattern DUALIZATION, OUTER LINEARIZATION/RELAXATION. In the context of (4.15), for example, one would dualize with respect to the constraints $\bar{A}x \leq \bar{b}$, outer-linearize the resulting minimand in the obvious way, and then apply Relaxation.

[Dantzig and Wolfe 60]

The well-known Dantzig-Wolfe decomposition approach for linear programs will be explained in terms of the linear program

$$\text{(4.15)} \qquad \underset{x \geq 0}{\text{Maximize}}\ c^t x \ \text{s.t.}\ \begin{array}{l} Ax \leq b, \\ \bar{A}x \leq \bar{b}, \end{array}$$

where we have arbitrarily divided the constraints into two groups. With the definition

$$\text{(4.16)} \qquad X \triangleq \{x \geq 0 : Ax \leq b\},$$

we may write (4.15) as

$$\text{(4.17)} \qquad \underset{x \in X}{\text{Maximize}}\ c^t x \ \text{s.t.}\ \bar{A}x \leq \bar{b}.$$

Since X is a convex polytope, we know (Th. 3) that it admits an exact inner linearization using only a finite number of points. Invoking this representation for X, we obtain a master linear program with a vast number of variables to which Restriction can be applied in the form of the Simplex Method. It turns out that the pricing operation (cf. Sec. 3.2) can be accomplished by solving a linear subproblem whose feasible region is X. The details are as follows.

Assume that X is not empty and also, for ease of exposition only, that X is bounded. Then X can be represented in terms of its extreme points $\langle x^1, \ldots, x^p \rangle$, and (4.17) can be written as the equivalent master linear program

$$(4.18) \qquad \underset{\alpha \geqq 0}{\text{Maximize}}\ c^t\left(\sum_{j=1}^{p} \alpha_j x^j\right) \text{ s.t. } \sum_{j=1}^{p} \alpha_j = 1,$$
$$\bar{A}\left(\sum_{j=1}^{p} \alpha_j x^j\right) \leqq \bar{b}.$$

The Simplex Method for this problem corresponds to Restriction with respect to the constraints $\alpha \geqq 0$.† To describe how the pricing operation can be mechanized, we shall use the familiar terminology of linear programming rather than the general terminology of Restriction. The optimality conditions at the general iteration are $u \geqq 0$ and

$$(4.19) \qquad u_o + u^t \bar{A} x^j - c^t x^j \geq 0,\ j = 1, \ldots, p,$$

†Actually, the inequality constraints involving $\bar{A}$ are also normally considered as candidates for restriction to equality. The latter constraints can be excluded, if desired, from the candidates for restriction by giving $u \geqq 0$ priority over (4.19) in determining the entering basic variable. Such a modification is necessary, as we shall see later in this subsection, when nonlinear functions are inner-linearized.

where u_o and the vector u are the current Simplex multipliers. Condition (4.19) is equivalent to

$$[u_o + \underset{1\le j\le p}{\text{Minimum}} \{(u^t\bar{A} - c^t)x^j\}] \ge 0$$

or, since $<x^1, \ldots, x^p>$ span X, to

$$(4.20) \qquad [u_o + \underset{x\in X}{\text{Min}} (u^t\bar{A} - c^t)x] \ge 0.$$

The linear program in this expression is valid replacement for the finite minimum in the previous expression because the minimum of a linear function over X occurs at an extreme point. Thus we see how to test optimality when the Simplex Method is applied to (4.18). If either $u \geqq 0$ or (4.20) fails to hold, a profitable nonbasic variable satisfying the usual criterion for the entering variable is obtained automatically: if the greatest violation occurs in $u \geqq 0$, introduce the corresponding slack variable; if in (4.20), introduce the variable α_{j_o}, where x^{j_o} is an optimal basic feasible solution of the linear program in (4.20) (the extremal function coefficient of α_{j_o} is $c^tx^{j_o}$, and the technological coefficient column is unity followed by $\bar{A}x^{j_o}$).

Thus there is no difficulty in carrying out the Simplex Method applied to the master problem (4.18). Each iteration requires solving the linear subproblem in (4.20).† This approach may possess an advantage over the direct application of the Simplex Method to (4.15) when

---

†The subproblem need be solved from scratch only at the first iteration; thereafter, restarting or parametric techniques can be used to recover an optimum as u changes from iteration to iteration.

the subproblem has some special structure. For example, if (4.15) is a transportation problem with additional constraints, then the subproblem becomes a pure transportation problem if $\bar{A}$ is taken to comprise the additional constraints. Another example is the case in which A is block-diagonal, for then the subproblem separates into k independent smaller linear programs. In general, one should select a grouping of the constraints (in terms of A and $\bar{A}$) that isolates a special structure, and then exploit this structure in dealing with (4.20). See [Broise, Huard and Sentenac 68], [Orchard-Hays 68, Sec. 10.4] for additional discussion based on computational experience.

[Dantzig 63a, Ch. 24]

Now consider a nonlinear version of (4.17), namely

$$(4.21) \qquad \underset{x \in X}{\text{Maximize}}\ f(x) \text{ s.t. } g_i(x) \le b_i,\ i = 1, \ldots, m,$$

where X is a convex set, f is concave on X, and g_i is convex on X. Dantzig and Wolfe's approach [Dantzig 63a, Ch. 24] for this problem can be viewed as follows. Let f and each g_i be approximated by Inner Linearization over an arbitrarily fine base $\langle x^1, x^2, \ldots \rangle$ in X, so that (4.21) is approximated as closely as desired (in principle, at least) by the linear master problem

$$(4.22) \qquad \underset{\alpha \geq 0}{\text{Maximize}}\ \Sigma_j\ \alpha_j f(x^j) \text{ s.t. } \Sigma_j\ \alpha_j = 1,$$
$$\Sigma_j\ \alpha_j g_i(x^j) \le b_i,\ i = 1, \ldots, m.$$

We say "in principle" because we do not wish to actually evaluate f and each g_i at every point in the base, or even specify the base explicitly. Hence it is natural to solve (4.22) by Restriction with

the constraints $\alpha \geq 0$ as the candidates for restriction to equality (when α_j is restricted to 0, the values $f(x^j)$ and $g_i(x^j)$ are not needed). A very natural way to do this is to employ the Simplex Method with a priority convention to ensure that the restricted problems are truly optimized: slack variables corresponding to the g_i constraints must be given priority over structural variables in determining which variable is to enter a basis. Any feasible solution of (4.21) can be used to find an initial basic feasible solution, and at the general iteration the optimality criterion or pricing problem is (cf. (4.19)) $u_i \geq 0$ $(1 \leq i \leq m)$ and

$$(4.23) \qquad u_o + \sum_{i=1}^{m} u_i g_i(x^j) - f(x^j) \geq 0, \text{ all } j,$$

where $u_o, u_1, \ldots, u_m$ are the current Simplex multipliers. By the priority convention, we may assume that $u_i \geq 0$ $(1 \leq i \leq m)$. Note that (4.23) is intimately related (cf. (4.20)) to the convex subproblem

$$(4.24) \qquad \underset{x \in X}{\text{Minimize}} \sum_{i=1}^{m} u_i g_i(x) - \imath \quad).$$

If u_o plus the optimal value of this problem is nonnegative, then (4.23) holds and an optimal solution of (4.21) is at hand $[x^* = \Sigma_j \hat{\alpha}_j x^j$, where $\hat{\alpha}$ is the current and optimal solution of (4.22)]; otherwise, an optimal or near-optimal solution $\hat{x}$ of (4.24) can be profitably added to the current explicit base by introducing the corresponding α_j into the basis in the usual way after evaluating $f(\hat{x})$ and $g_i(\hat{x})$. In practice, termination would take place as soon as the value of the current approximation to an optimal solution of

(4.21)--the quantity $f(\Sigma_j \hat{\alpha}_j x_j)$--approaches closely enough the following easily demonstrated upper bound for the true optimal value:

$$(4.25) \qquad \sum_{i=1}^{m} u_i b_i - \mathrm{Min}_{x \in X} \left[\sum_{i=1}^{m} u_i g_i(x) - f(x) \right].$$

This approach is particularly attractive when the structure is such that (4.24) is relatively tractable by comparison with (4.21); for example, when X is an open set and f and g_i are differentiable, or when (4.24) is separable into several independent subproblems.

A Variant

It is interesting to observe that Inner Linearization need not be applied to all nonlinear functions of (4.21).[†] An advantage can sometimes be gained by inner-linearizing only a subset of the nonlinear functions, say $g_1, \ldots, g_{m_1}$ $(m_1 < m)$. Then instead of (4.22) we have the concave master problem

$$(4.26) \qquad \begin{aligned} &\underset{\alpha \geq 0}{\text{Maximize}}\ f(\Sigma_j\ \alpha_j x^j) \\ &\text{s.t. } \Sigma_j\ \alpha_j = 1, \\ &\qquad \Sigma_j\ \alpha_j g_i(x^j) \leq b_i,\ i = 1, \ldots, m_1 \\ &\qquad g_i(\Sigma_j\ \alpha_j x^j) \leq b_i,\ i = m_1 + 1, \ldots, m. \end{aligned}$$

Again we wish to apply Restriction with only the nonnegativity constraints $\alpha \geq 0$ as candidates for restriction to equality. The Simplex Method can no longer be adapted to this purpose, however, since (4.26) is not a linear program. Implementation requires a concave programming

---

[†] In [Whinston 66], for example, the objective function of a block-diagonal quadratic program with coupling constraints is not inner-linearized.

algorithm for solving the restricted versions of (4.26) and also a means of mechanizing the pricing operation. We need not discuss the first requirement. The second involves being able to determine the prices μ_j^s for all j in S, where S is the current set of indices for which α_j is restricted to value 0. This can be done as follows [Holloway 69]. Let α^s be the optimal solution to (4.26) with the additional restrictions $\alpha_j = 0$ for $j\varepsilon S$, and let $u_o^s, u_1^s, \ldots, u_m^s$ be the associated optimal multipliers (which must exist if a constraint qualification is satisfied). Then, assuming all functions are continuously differentiable, the price μ_j^s associated with $\alpha_j = 0$ is given for all $j\varepsilon S$ by

$$(4.27)\quad \mu_j^s = u_o^s - \nabla f(x^s)x^j + \sum_{i=1}^{m_1} u_i^s g_i(x^j) + \sum_{i=m_1+1}^{m} u_i^s \nabla g_i(x^s)x_j ,$$

where

$$(4.28)\qquad x^s \triangleq \sum_{j\notin S} \alpha_j^s x^j .$$

It follows that the pricing problem can be solved by optimizing the convex ($u_i^s \geq 0$) subproblem

$$(4.29)\quad \underset{x\varepsilon X}{\text{Minimize}} \; -\nabla f(x^s)x + \sum_{i=1}^{m_1} u_i^s g_i(x) + \sum_{i=m_1+1}^{m} u_i^s \nabla g_i(x^s)x.$$

Compare with (4.24). If f were inner-linearized too, the first term of the maximand of (4.29) would be -f(x).

Which of all given constraints should be incorporated into X,

and which of the remainder and whether f itself should be inner-linearized, depends mainly on the availability of efficient algorithms for the resulting versions of (4.29) and (4.26) with $\alpha_j = 0$ for $j \in S$.

4.4 [Geoffrion 68b, Sec. 4]

A quite general problem with multidivisional structure is

$$(4.30)\qquad \begin{aligned} &\underset{x}{\text{Maximize}} \sum_{i=1}^{k} f_i(x_i) \\ &\text{s.t.} \quad H_i(x_i) \geqq 0,\ i = 1, \ldots, k \\ &\qquad\quad \sum_{i=1}^{k} G_i(x_i) \geqq b, \end{aligned}$$

where f_i, h_{ij} and g_{ij} are all concave differentiable functions of the vector x_i. The subscript i can be thought of as indexing the individual divisions, which are linked together only by coupling constraints. The approach of [Geoffrion 68b, Sec. 4] is an application of the pattern **PROJECTION/FEASIBLE DIRECTIONS.** The optimization of (4.30) is carried out largely at the divisional level subject to central coordination.

First (4.30) is projected onto the space of its coupling constraints. This requires introducing the vectors $y_1, \ldots, y_k$:

$$(4.31)\qquad \begin{aligned} &\underset{x,y}{\text{Maximize}} \sum_{i=1}^{k} f_i(x_i) \\ &\text{s.t.} \quad H_i(x_i) \geqq 0,\ i = 1, \ldots, k \\ &\qquad\quad G_i(x_i) \geqq y_i,\ i = 1, \ldots, k \\ &\qquad\quad \sum_{i=1}^{k} y_i \geqq b. \end{aligned}$$

In effect, this changes the given problem from one with coupling constraints to one with coupling variables, since (4.31) separates into k separate problems if y is held fixed temporarily. One may interpret y_i as a vector of resources and tasks assigned to the i^{th} division. Projection of this problem onto y yields the master problem

$$\text{(4.32)} \qquad \underset{y}{\text{Maximize}} \sum_{i=1}^{k} v_i(y_i) \text{ s.t. } \sum_{i=1}^{k} y_i \geqq b,$$

where v_i is defined as the supremal value of the parameterized divisional problem

$$\text{(4.33)} \qquad \underset{x_i}{\text{Maximize}}\ f_i(x_i) \text{ s.t. } \begin{array}{l} H_i(x_i) \geqq 0, \\ G_i(x_i) \geqq y_i. \end{array}$$

Now we wish to apply the Feasible Directions strategy to (4.32). The idea of this strategy, it will be recalled, is to generate an improving sequence of feasible points, with each new point determined from the previous one by selecting an improving feasible direction and then maximizing along a line emanating in this direction. The latter maximization is only one-dimensional, and can easily be essentially decentralized to the divisional level. The chief difficulty with this strategy concerns how to find a good improving feasible direction, for the maximand $\sum_{i=1}^{k} v_i(y_i)$ is not everywhere differentiable and is available only implicitly in terms of the divisional problems (4.33). It can nevertheless be shown [ibid., Sec. 4.2], using the theory of subgradients for concave functions and the optimality conditions associated with (4.33), that the following explicit linear program yields

an improving feasible direction z^o for (4.32) at a feasible point y^o; moreover, z^o is best among all feasible directions in that it maximizes the initial rate of improvement of $\sum_{i=1}^{k} v_i(y_i)$:

$$(4.34) \qquad \begin{aligned} &\underset{w,z}{\text{Maximize}} \ \sum_{i=1}^{k} \nabla f_i^o w_i \\ &\text{s.t.} \quad \nabla g_{ij}^o w_i - z_{ij} \geq 0, \ i = 1, \ldots, k \\ &\qquad\qquad\qquad\qquad j \text{ such that } g_{ij}^o = y_{ij}^o \\ &\qquad \nabla h_{ij}^o w_i \geq 0, \ i = 1, \ldots, k \\ &\qquad\qquad\qquad\qquad j \text{ such that } h_{ij}^o = 0 \\ &\qquad \sum_{i=1}^{k} z_{ij} \geq 0, \ j \text{ such that } \sum_{i=1}^{k} y_{ij}^o = b_j \\ &\qquad -1 \leq z_{ij} \leq 1, \text{ all } i \text{ and } j. \end{aligned}$$

Here ∇g_{ij}^o refers to a row vector that is the gradient of g_{ij} evaluated at an optimal solution of (4.33) with $y_i = y_i^o$, and the other superscripted quantities have similar definitions. The vector w_i has the same dimension as x_i. This subproblem enables the Feasible Directions strategy for (4.32) to be carried out.

4.5 [Takahashi 64]

Consider

$$(4.35) \qquad \underset{x}{\text{Maximize}} \ f(x) \ \text{s.t.} \ H(x) = 0, \quad G(x) = 0,$$

where f is concave and all constraints are linear. Suppose that the G constraints are *complicating* in the sense that the problem would be much easier if they were not present. For instance, the complicating

constraints may be the coupling constraints of a structure similar to the one in the previous subsection, or they may spoil what would otherwise be a special structure for which efficient solution methods would be available. The pattern of the "local" approach of [Takahashi 64] for this problem is DUALIZATION/FEASIBLE DIRECTIONS.

The dual of (4.35) with respect to the complicating constraints only yields (see, e.g., [Rockafellar 68] or [Geoffrion 69]) the following problem in the space of the dual variables λ (a vector whose dimension matches G):

$$\text{(4.36)} \qquad \underset{\lambda}{\text{Minimize}}\ v(\lambda),$$

where $v(\lambda)$ is defined as the supremal value of the parameterized problem

$$\text{(4.37)} \qquad \underset{x}{\text{Maximize}}\ f(x) + \lambda^t G(x) \text{ s.t. } H(x) = 0.$$

Note that (4.37) is of the same form as (4.35) except the complicating constraints are now part of the criterion function.

To apply the Feasible Directions strategy to (4.36), we must be able to identify an improving feasible direction. Any direction is feasible, of course, since λ is unconstrained. When f is strictly concave, it can be shown that v is differentiable. Its gradient at a point λ^o is simply $G(x^o)$, where x^o is the optimal solution of (4.37) with $\lambda = \lambda^o$. Hence the Feasible Directions strategy can be carried out for (4.36) using the negative of the gradient of v

as the improving feasible direction. Actually, Takahashi proposes a short-step method rather than requiring a one-dimensional minimization to be performed in order to determine step size. The procedure may be summarized as follows.

1. Choose a starting point λ^o.

2. Solve (4.37) with $\lambda = \lambda^o$ for its optimal solution x^o. If $G(x^o) = 0$, then x^o is optimal in (4.35); otherwise, go on to Step 3.

3. Let $\lambda' = \lambda^o - \zeta G(x^o)$, where ζ is a small positive constant, and return to Step 2 with λ' in place of λ^o.

5. CONCLUSION

We have attempted to develop a framework of unifying concepts that comprehends much of the literature on large-scale mathematical programming. If we have been successful, the non-specialist should have an overview of the field that facilitates further study, and the advanced reader should feel that he has a deeper understanding of previously familiar algorithms and that he perceives new commonalities among approaches that heretofore seemed to be related only vaguely if at all.

In addition, we hope that the framework will suggest a variety of worthwhile topics for investigation. The problem manipulations and solution strategies discussed here all invite further study, and others should be added to the fold so that additional algorithms can be encompassed. The algorithms falling within the purview of each particular manipulation/strategy pattern (cf. Table 2) should be studied carefully in relation to one another, with the aim of learning how "best" to use the tactical options of the pattern and organize the computations for various classes of problems.

The relationships between ostensibly different patterns also warrant further study. We mentioned in Sec. 3.3 that Restriction (Relaxation) is essentially equivalent to Dualization followed by Relaxation (Restriction), and other equivalences were briefly noted in Secs. 4.2 and 4.3. Many others exist; for example, it has often been observed that Dantzig-Wolfe and Benders Decomposition are dual to one another in an appropriate sense. The results of [Zoutendijk 60; Secs. 9.4, 10.3, 11.4] are in this spirit, even if they do not specifically involve algorithms for large-scale programming. Knowledge of such

relations reduces the number of essentially different patterns to be considered, and enables meaningful comparisons among the remainder.

Investigations along these lines should help civilize the jungle of extant algorithms and pave the way for truly significant computational studies.

BIBLIOGRAPHY[†]

Abadie, J., and M. Sakarovitch, 1967. "Two Methods of Decomposition for Linear Programs," presented at the International Symposium on Mathematical Programming, Princeton, New Jersey, August 14-18.

Abadie, J., and A. C. Williams, 1963. "Dual and Parametric Methods in Decomposition," in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

Aoki, M., 1968. "Planning Procedure Under Increasing Returns," Harvard Economic Research Project, Harvard University, Cambridge.

Appelgren, L. H., 1969. "A Column-Generation Algorithm for a Ship Scheduling Problem," Transportation Science, 3, 1 (February), pp. 53-68.

Bakes, M. D., 1966. "Solution of Special Linear Programming Problems with Additional Constraints," Operational Research Quarterly, 17, 4 (December), pp. 425-445.

Balas, E., 1965. "Solution of Large Scale Transportation Problems Through Aggregation," Operations Research, 13, 1 (January-February), pp. 82-93.

Balas, E., 1966. "An Infeasibility-Pricing Decomposition Method for Linear Programs," Operations Research, 14, 5 (September-October), pp. 847-873.

Balinski, M. L., 1964. "On Some Decomposition Approaches in Linear Programming," in Recent Mathematical Advances in Operations Research, Engineering Summer Conferences, University of Michigan.

Balinski, M. L., and P. Wolfe, 1963. "On Benders Decomposition and a Plant Location Problem," Working Paper ARO-27 (December), Mathematica, Princeton, New Jersey.

Baumol, W. J., and T. Fabian, 1964. "Decomposition, Pricing for Decentralization and External Economies," Management Science, 11, 1 (September), pp. 1-32.

Beale, E. M. L., 1963. "The Simplex Method Using Pseudo-basic Variables for Structured Linear Programs," in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

Beale, E. M. L., 1967. "Decomposition and Partitioning Methods for Nonlinear Programming," in J. Abadie (ed.), Non-Linear Programming, North-Holland Publishing Company, Amsterdam, pp. 197-205.

Bell, E. J., 1966. "Primal-Dual Decomposition Programming," Preprints of the Proceedings of the Fourth International Conference on Operational Research, International Federation of Operational Research Societies, Boston.

Bellman, R., 1957. "On the Computational Solution of Linear Programming Problems Involving Almost-Block-Diagonal Matrices," Management Science, 3, 4 (July), pp. 403-406.

[†]Items marked * are not ordinarily considered part of the large-scale mathematical programming literature.

Benders, J. F., 1962. "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik, 4, pp. 238-252.

Bennett, J. M., 1966. "An Approach to Some Structured Linear Programming Problems," Operations Research, 14, 4 (July-August), pp. 636-645.

Bennett, J. M., and D. R. Green, 1966. "A Method for Solving Partially Linked Linear Programming Problems," Basser Computing Department Technical Report No. 38 (February), University of Sydney.

*Berge, C., 1963. Topological Spaces, Macmillan Company, New York.

*Berge, C., and A. Ghouila-Houri, 1965. Programming, Games and Transportation Networks, John Wiley and Sons, New York.

Bessiere, F., and E. A. Sautter, 1968. "Optimization and Suboptimization: The Method of Extended Models in the Nonlinear Case," Management Science, 15, 1 (September), pp. 1-11.

Bradley, S. P., 1965. "Solution Techniques for the Traffic Assignment Problem," Operations Research Center Report 65-35 (June), University of California, Berkeley.

Bradley, S. P., 1967. "Decomposition Programming and Economic Planning," Operations Research Center Report 67-20 (June), University of California, Berkeley.

Bradley, S. P., 1968. "Nonlinear Programming Via the Conjugate Dual," IBM Report (June), Federal Systems Center, Gaithersburg, Maryland.

Broise, P., P. Huard, and J. Sentenac, 1968. "Décomposition des programmes mathématiques," Monographies de Recherche Opérationnelle, No. 6, Dunod, Paris.

Buchet, J. de, 1966. "Expériences et statistiques sur la résolution des programmes linéaires de grandes dimensions," reprinted in D. B. Hertz and J. Melese (eds.), Proceedings of the Fourth International Conference on Operational Research, John Wiley and Sons, New York, 1969.

Buzby, B. R., B. J. Stone, and R. L. Taylor, 1965. "Computational Experience with a Nonlinear Distribution Program," privately communicated to M. L. Balinski (January). See M. L. Balinski, "Integer Programming: Methods, Uses, Computation," Management Science, 12, 3 (November 1965), p. 308.

Chandy, K. M., 1968. "Parametric Primal Partitioning," Operations Research Center, Massachusetts Institute of Technology, Cambridge.

Charnes, A., R. W. Clower, and K. O. Kortanek, 1967. "Effective Control Through Coherent Decentralization with Preemptive Goals." Econometrica, 35, 2 (April), pp. 294-320.

Charnes, A., and W. W. Cooper, 1955. "Generalizations of the Warehousing Model." Operational Research Quarterly, 6, 4 (December), pp. 131-172.

Charnes, A., W. W. Cooper, and K. O. Kortanek, 1969. "On the Theory of Semi-Infinite Programming and a Generalization of the Kuhn-Tucker Saddle Point Theorem for Arbitrary Convex Functions," Naval Research Logistics Quarterly, 16, 1 (March), pp. 41-51.

Charnes, A., and K. O. Kortanek, 1968. "On the Status of Separability and Non-Separability in Decentralization Theory," Management Science, 15, 2 (October), pp. 12-14.

*Charnes, A., and C. Lemke, 1954. "Minimization of Nonlinear Separable Convex Functionals," Naval Research Logistics Quarterly, 1, pp. 301-312.

*Cheney, E. W., and A. A. Goldstein, 1959. "Newton's Method for Convex Programming and Tchebycheff Approximation," Numerische Mathematik, 1, pp. 253-268.

Cobb, R. H., and J. Cord, 1967. "Decomposition Approaches for Solving Linked Programs," presented at the International Symposium on Mathematical Programming, Princeton, New Jersey, August 14-18.

Dantzig, G. B., 1955a. "Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming," Econometrica, 23, 2 (April), pp. 174-183.

Dantzig, G. B., 1955b. "Optimal Solution of a Dynamic Leontief Model with Substitution," Econometrica, 23, 3 (July), pp. 295-302.

Dantzig, G. B., 1959. "On the Status of Multistage Linear Programming Problems," Management Science, 6, 1 (October), pp. 53-72.

Dantzig, G. B., 1960. "A Machine-Job Scheduling Model," Management Science, 6, 2 (January), pp. 191-195.

Dantzig, G. B., 1963a. Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey.

Dantzig, G. B., 1963b. "Compact Basis Triangularization for the Simplex Method," in R. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

Dantzig, G. B., 1966. "Linear Control Processes and Mathematical Programming," SIAM Journal on Control, 4, 1 (February), pp. 56-60.

Dantzig, G. B., 1968. "Large-Scale Linear Programming," in G. B. Dantzig and A. F. Veinott, Jr. (eds.), Mathematics of the Decision Sciences, Part 1, American Mathematical Society, Providence.

Dantzig, G. B., W. Blattner, and M. R. Rao, 1967. "Finding a Cycle in a Graph with Minimum Cost to Time Ratio with Applications to a Ship Routing Problem," Theory of Graphs International Symposium, Dunod, Paris, pp. 77-83.

Dantzig, G. B., D. R. Fulkerson, and S. Johnson, 1954. "Solution of a Large-Scale Traveling Salesman Problem," Operations Research, 2, 4 (November), pp. 393-410.

Dantzig, G. B., R. P. Harvey and R. D. McKnight, 1964. "Updating the Product Form of the Inverse for the Revised Simplex Method," Operations Research Center Report 64-33 (December), University of California, Berkeley.

Dantzig, G. B., and D. L. Johnson, 1964. "Maximum Payload Per Unit Time Delivered Through an Air Network," Operations Research, 12, 2 (March-April), pp. 230-236.

Dantzig, G. B., and A. Madansky, 1961. "On the Solution of Two-Stage Linear Programs Under Uncertainty," in Proceedings of the Fourth Symposium on Mathematical Statistics and Probability, I, University of California Press, Berkeley, pp. 165-176.

Dantzig, G. B., and W. Orchard-Hays, 1954. "The Product Form of the Inverse in the Simplex Method," Mathematical Tables and Aids to Computation, 8, pp. 64-67.

Dantzig, G. B., and R. M. Van Slyke, 1967. "Generalized Upper Bounded Techniques for Linear Programming," Journal of Computer and System Sciences, 1, 3 (October), pp. 213-226.

Dantzig, G. B., and P. Wolfe, 1960. "Decomposition Principle for Linear Programs," Operations Research, 8, 1 (January-February), pp. 101-111. See also Econometrica, 29, 4 (October), pp. 767-778.

Dzielinski, B. P., and R. E. Gomory, 1965. "Optimal Programming of Lot Sizes, Inventory and Labor Allocations," Management Science, 11, 9 (July), pp. 874-890.

El Agizy, M., 1967. "Two-Stage Programming Under Uncertainty with Discrete Distribution Function," Operations Research, 15, 1 (January-February), pp. 55-70.

Elmaghraby, S. E., 1968. "A Loading Problem in Process Type Production," Operations Research, 16, 5 (September-October), pp. 902-914.

*Falk, J., 1965. "An Algorithm for Separable Convex Programming under Linear Equality Constraints," Research Analysis Corporation Technical Paper 148 (March), McLean, Virginia.

*Falk, J., 1967. "Lagrange Multipliers and Nonlinear Programming," Journal of Mathematical Analysis and Application, 19, 1, pp. 141-159.

*Fiacco, A. V., and G. P. McCormick, 1968. Nonlinear Programming, John Wiley and Sons, New York.

Ford, L. R., and D. R. Fulkerson, 1958. "A Suggested Computation for Maximal Multi-Commodity Network Flows," Management Science, 5, 1 (October), pp. 97-101.

Fox, B., 1969a. "Finding Minimal Cost-Time Ratio Circuits," Operations Research, 17, 3 (May-June), pp. 546-551.

Fox, B., 1969b. Stability of the Dual Cutting-Plane Algorithm for Concave Programming, The RAND Corporation, RM-6147-PR, forthcoming.

Gass, S. I., 1966. "The Dualplex Method for Large-Scale Linear Programs," Operations Research Center Report 66-15 (June), University of California, Berkeley.

Geoffrion, A. M., 1966. "A Markovian Procedure for Concave Programming with Some Linear Constraints," reprinted in D. B. Hertz and J. Melese (eds.), Proceedings of the Fourth International Conference on Operational Research, John Wiley and Sons, New York, 1969.

*Geoffrion, A. M., 1967. "Reducing Concave Programs with Some Linear Constraints," SIAM Journal on Applied Mathematics, 15, 3 (May), pp. 653-664.

Geoffrion, A. M., 1968a. "Relaxation and the Dual Method in Mathematical Programming," Western Management Science Institute Working Paper No. 135 (March), University of California, Los Angeles.

Geoffrion, A. M., 1968b. Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems, The RAND Corporation, RM-5829-PR (December).

Geoffrion, A. M., 1969. "Duality in Nonlinear Programming: A Simplified Applications-Oriented Development," Western Management Science Institute Working Paper No. 150 (August), University of California, Los Angeles.

Gilmore, P. C., and R. E. Gomory, 1961. "A Linear Programming Approach to the Cutting-Stock Problem," Operations Research, 9, 6 (November-December), pp. 849-859.

Gilmore, P. C., and R. E. Gomory, 1963. "A Linear Programming Approach to the Cutting-Stock Problem--Part II," Operations Research, 11, 6 (November-December), pp. 863-888.

Gilmore, P. C., and R. E. Gomory, 1965. "Multistage Cutting Stock Problems of Two and More Dimensions," Operations Research, 13, 1 (January-February), pp. 94-120.

Glassey, C. R., 1966. "An Algorithm for a Machine Loading Problem," Operations Research Center Working Paper 30 (September), University of California, Berkeley.

Glassey, C. R., 1968. "Dynamic Linear Programs for Production Scheduling," Operations Research Center Report 68-2 (January), University of California, Berkeley.

*Goldman, A. J., 1956. "Resolution and Separation Theorems for Polyhedral Convex Sets," in H. W. Kuhn and A. W. Tucker (eds.), Linear Inequalities and Related Systems, Princeton University Press, Princeton, New Jersey.

Golshtein, E. G., 1966. "A General Approach to the Linear Programming of Block Structures," Soviet Physics--Doklady, 11, 2 (August), pp. 100-103.

*Gomory, R. E., 1958. "An Algorithm for Integer Solutions to Linear Programs," reprinted in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York, 1963.

Gomory, R. E., 1963. "Large and Nonconvex Problems in Linear Programming," in Proceedings of Symposia in Applied Mathematics, XV, American Mathematical Society, pp. 125-139.

Gomory, R. E., and T. C. Hu, 1962. "An Application of Generalized Linear Programming to Network Flows," Journal of the Society for Industrial and Applied Mathematics, 10, 2 (June) pp. 260-283.

Gomory, R. E., and T. C. Hu, 1964. "Synthesis of a Communication Network," Journal of the Society for Industrial and Applied Mathematics, 12, 2 (June), pp. 348-369.

Gould, S., 1959. "A Method of Dealing with Certain Non-Linear Allocation Problems Using the Transportation Technique," Operational Research Quarterly, 10, 3 (September), pp. 138-171.

Graves, G. W., G. B. Hatfield, and A. Whinston, 1969. "Water Pollution Control Using By-Pass Piping," Water Resources Research, 5, 1 (February).

Grigoriadis, M. D., 1969. "A Dual Generalized Upper Bounding Technique," IBM New York Scientific Center Report No. 320-2973 (June), New York.

Grigoriadis, M. D., and K. Ritter, 1968. "A Decomposition Method for Structured Linear and Non-Linear Programs," Computer Sciences Technical Report No. 10 (January), University of Wisconsin, Madison.

Grigoriadis, M. D., and W. F. Walker, 1968. "A Treatment of Transportation Problems by Primal Partition Programming," Management Science, 14, 9 (May), pp. 565-599.

*Grinold, R. C., 1968. "Continuous Programming," Operations Research Center Report 68-14 (June), University of California, Berkeley.

Grinold, R. C., 1969. "Steepest Ascent for Large-Scale Linear Programs," Technical Report No. 587 (April), Division of Engineering and Applied Physics, Harvard University, Cambridge.

*Hartley, H. O., and R. R. Hocking, 1963. "Convex Programming by Tangential Approximation," Management Science, 9, 4 (July), pp. 600-612.

Hass, J. E., 1968. "Transfer Pricing in a Decentralized Firm," Management Science, 14, 6 (February), pp. B-310-B-331.

Heesterman, A. R. G., and J. Sandee, 1965. "Special Simplex Algorithm for Linked Problems," Management Science, 11, 3 (January), pp. 420-428.

Holloway, C. A., 1969. "A Mathematical Programming Approach to Decision Processes for Complex Operational Systems, With the Aggregate Planning Problem as an Example," Ph.D. dissertation, Graduate School of Business Administration, University of California, Los Angeles.

Hopkins, D. S. P., 1969. "Sufficient Conditions for Optimality in Infinite Horizon Linear Economic Models," Operations Research House Technical Report 69-3 (March), Stanford University.

*Joksch, H. C., 1964. "Programming with Fractional Linear Objective Functions," Naval Research Logistics Quarterly, 11, 2 & 3 (June-September), pp. 197-204.

Kaul, R. N., 1965a. "Comment on 'Generalized Upper Bounded Techniques in Linear Programming'," Operations Research Center Report 65-21 (August), University of California, Berkeley.

Kaul, R. N., 1965b. "An Extension of Generalized Upper Bounded Techniques for Linear Programming," Operations Research Center Report 65-27 (August), University of California, Berkeley.

*Kelley, J. E., Jr., 1960. "The Cutting-Plane Method for Solving Convex Programs," Journal of the Society for Industrial and Applied Mathematics, 8, 4 (December), pp. 703-712.

*Kohler, D. A., 1967. "Projections of Convex Polyhedral Sets," Operations Research Center Report 67-29 (August), University of California, Berkeley.

Kornai, J., and T. Liptak, 1965. "Two-Level Planning," Econometrica, 33, 1 (January), pp. 141-169.

Kronsjö, T. O. M., 1968a. "Centralization and Decentralization of Decision Making," Revue d'Informatique et de Recherche Opérationelle, 2, 10, pp. 73-114.

Kronsjö, T. O. M., 1968b. "Optimal Coordination of a Large Convex Economic System," Discussion Paper Series RC/A, No. 9 (January), University of Birmingham, Faculty of Commerce and Social Sciences, Birmingham, Great Britain.

Kunzi, H. P., and S. T. Tan, 1966. Lineare Optimierung Grosser Systeme, Springer-Verlag, Berlin.

Lasdon, L. S., 1964. "A Multi-Level Technique for Optimization," Systems Research Center Report 50-64-19, Case Institute of Technology, Cleveland.

Lasdon, L. S., 1968. "Duality and Decomposition in Mathematical Programming," IEEE Transactions on System Science and Cybernetics, SSC-4, 2 (July), pp. 86-100.

Lasdon, L. S., 1969. Optimization Theory for Large Systems, The Macmillan Company, New York, forthcoming.

Lasdon, L. S., and J. E. Mackey, 1968. "An Efficient Algorithm for Multi-Item Scheduling," Technical Memorandum No. 108 (May), Department of Operations Research, Case Western Reserve University, Cleveland.

Lasdon, L. S., and J. D. Schoeffler, 1966. "Decentralized Plant Control," I.S.A. Transactions, 5, April, pp. 175-183.

Madge, D. N., 1965. "Decomposition of Mine Scheduling Problems Involving Mining Sequence Restrictions," C.O.R.S. Journal, 3, 3 (November), pp. 161-165.

Malinvaud, E., 1967. "Decentralized Procedures for Planning," in E. Malinvaud and M.O.L. Bacharach (eds.), Activity Analysis in the Theory of Economic Growth, Macmillan Company, New York.

Manne, A. S., and H. M. Markowitz (eds.), 1963. Studies in Process Analysis, John Wiley and Sons, New York.

Markowitz, H., 1957. "The Elimination Form of the Inverse and Its Application to Linear Programming," Management Science, 3, 3 (April), pp. 255-269.

Metz, C. K. C., R. N. Howard, and J. M. Williamson, 1966. "Applying Benders Partitioning Method to a Non-Convex Programming Problem," reprinted in D. B. Hertz and J. Melese (eds.), Proceedings of the Fourth International Conference on Operational Research, John Wiley and Sons, New York, 1969.

Midler, J. L., and R. D. Wollmer, 1968. A Flight Planning Model for the Military Airlift Command, The RAND Corporation, RM-5722-PR, October.

*Miller, C. E., 1963. "The Simplex Method for Local Separable Programming," in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

Nemhauser, G. L., 1964. "Decomposition of Linear Programs by Dynamic Programming," Naval Research Logistics Quarterly, 11, 2 & 3 (June-September), pp. 191-196.

Orchard-Hays, W., 1968. Advanced Linear Programming Computing Techniques, McGraw-Hill Book Company, New York.

*Orden, A., and V. Nalbandian, 1968. "A Bidirectional Simplex Algorithm," Journal of the Association for Computing Machinery, 15, 2 (April), pp. 221-235.

*Parikh, S. C., 1967. "Generalized Stochastic Programs with Deterministic Recourse," Operations Research Center Report 67-27 (July), University of California, Berkeley.

Parikh, S. C., and R. W. Shephard, 1967. "Linear Dynamic Decomposition Programming Approach to Long-Range Optimization of Northern California Water Resources System, Part I: Deterministic Hydrology," Operations Research Center Report 67-30 (August), University of California, Berkeley.

Pearson, J. D., 1966. "Decomposition, Coordination, and Multi-level Systems," IEEE Trans. on Systems Science and Cybernetics, SSC-2, 1 (August), pp. 36-40.

Rao, M. R., 1968. "Multi-Commodity Warehousing Models with Cash-Liquidity Constraints--A Decomposition Approach," Management Sciences Research Report No. 141 (September), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh.

Rao, M. R., and S. Zionts, 1968. "Allocation of Transportation Units to Alternative Trips--A Column Generation Scheme With Out-of-Kilter Subproblems," Operations Research, 16, 1 (January-February), pp. 52-63.

Rech, P., 1966. "Decomposition of Interconnected Leontief Systems by Square Block Triangularization," Pre-prints of the Proceedings of the Fourth International Conference on Operational Research, International Federation of Operational Research Societies, Boston.

*Ritter, K., 1967a. "A Parametric Method for Solving Certain Nonconcave Maximization Problems," Journal of Computer and System Sciences, 1, 1 (April), pp. 44-54.

Ritter, K., 1967b. "A Decomposition Method for Structured Quadratic Programming Problems," Journal of Computer and System Sciences, 1, 3 (October), pp. 241-260.

Ritter, K., 1967c. "A Decomposition Method for Linear Programming Problems with Coupling Constraints and Variables," Mathematics Research Center Technical Summary Report No. 739 (April), University of Wisconsin, Madison.

Robers, P. D., and A. Ben-Israel, 1968. "A Decomposition Method for Interval Linear Programming," Systems Research Memorandum No. 216 (August), The Technological Institute, Northwestern University, Evanston.

Robert, J. E., 1963. "A Method of Solving a Particular Type of Very Large Linear Programming Problem," C.O.R.S. Journal, 1, 1 (December), pp. 50-59.

*Rockafellar, R. T., 1968. "Duality in Nonlinear Programming," in G. B. Dantzig and A. F. Veinott, Jr. (eds.), Mathematics of the Decision Sciences, Part 1, American Mathematical Society, Providence.

Rosen, J. B., 1963. "Convex Partition Programming," in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

Rosen, J. B., 1964. "Primal Partition Programming for Block Diagonal Matrices," Numerische Mathematik, 6, pp. 250-260.

Rosen, J. B., 1967. "Optimal Control and Convex Programming," in J. Abadie (ed.), Nonlinear Programming, North-Holland Publishing Company, Amsterdam.

Rosen, J. B., and J. C. Ornea, 1963. "Solution of Nonlinear Programming Problems by Partitioning," Management Science, 10, 1 (October), pp. 160-173.

Saigal, R., 1966a. "Compact Basis Triangularization for the Block Angular Structures," Operations Research Center Report 66-1 (January), University of California, Berkeley.

Saigal, R., 1966b. "Block-Triangularization of Multi-Stage Linear Programs," Operations Research Center Report 66-9 (March), University of California, Berkeley. Revision issued as Working Paper No. 266 (April 1969), Center for Research in Management Science.

Sakarovitch, M., and R. Saigal, 1967. "An Extension of Generalized Upper Bounding Techniques for Structured LP Problems," SIAM Journal on Applied Mathematics, 15, 4 (July), pp. 906-914.

Silverman, G. J., 1968. "Primal Decomposition of Mathematical Programs by Resource Allocation," Technical Memorandum No. 116 (August), Operations Research Department, Case Western Reserve University, Cleveland.

*Stone, J. J., 1958. The Cross-Section Method: An Algorithm for Linear Programming, The RAND Corporation, P-1490 (September)

Takahashi, I., 1964. "Variable Separation Principle for Mathematical Programming," Journal of the Operations Research Society of Japan, 6, 1 (February), pp. 82-105.

Tcheng, T., 1966. "Scheduling of a Large Forestry-Cutting Problem by Linear Programming Decomposition," unpublished Ph.D. thesis, University of Iowa, August.

Thompson, G. L., F. M. Tonge, and S. Zionts, 1966. "Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems," Management Science, 12, 7 (March), pp. 588-608.

Tomlin, J. A., 1966. "Minimum-Cost Multicommodity Network Flows," Operations Research, 14, 1 (January-February), pp. 45-51.

*Uzawa, H., 1958. "Iterative Methods for Concave Programming," Chap. 10 in K. Arrow, L. Hurwicz and H. Uzawa (eds.), Studies in Linear and Nonlinear Programming, Stanford University Press, Stanford.

Van Slyke, R. M., 1968. "Mathematical Programming and Optimal Control," Operations Research Center Report 68-21 (July), University of California, Berkeley.

Van Slyke, R. M., and R. J. B. Wets, 1966. L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming," Operations Research Center Report 66-17 (July), revised June 1967, University of California, Berkeley.

Varaiya, P., 1966. "Decomposition of Large-Scale Systems," SIAM Journal on Control, 4, 1 (February), pp. 173-178.

*Veinott, A. F., Jr., 1967. "The Supporting Hyperplane Method for Unimodal Programming," Operations Research, 15, 1 (January-February), pp. 147-152.

Wagner, H. M., 1957. "A Linear Programming Solution to Dynamic Leontief Type Models," Management Science, 3, 3 (April), pp. 234-254.

Webber, D. W., and W. W. White, 1968. "An Algorithm for Solving Large Structured Linear Programming Problems," IBM New York Scientific Center Report No. 320-2946 (April), New York.

Weil, R. L., Jr., and P. C. Kettler, 1968. "Transforming Matrices to Use the Decomposition Algorithm for Linear Programs," Report 6801 (January), Department of Economics and Graduate School of Business, University of Chicago.

Weitzman, M., 1967. "Iterative Multi-Level Planning with Production Targets," Cowles Foundation Discussion Paper No. 239 (November), Yale University, New Haven. To appear in Econometrica.

Whinston, A., 1964. "A Dual Decomposition Algorithm for Quadratic Programming," Cahiers du Centre d'Etudes de Recherche Opérationnelle, 6, 4, pp. 188-201.

Whinston, A., 1966. "A Decomposition Algoritnm for Quadratic Program ming," Cahiers du Centre d'Etudes de Recherche Opérationnelle, 8, 2, pp. 112-131.

Williams, A. C., 1962. "A Treatment of Transportation Problems by Decomposition," Journal of the Society for Industrial and Applied Mathematics,"10, 1 (March), pp. 35-48.

Willoughby, R. A. (ed.), 1969. Proceedings of a Symposium on Sparse Matrices and Their Applications (Sept. 9-10, 1968), IBM Watson Research Center, Yorktown Heights, New York.

Wilson, R., 1966. "Computation of Optimal Controls," Journal of Mathematical Analysis and Application, 14, 1 (April), pp. 77-82.

Wismer, D. A. (ed.), 1969. Optimization Methods for Large-Scale Systems, forthcoming.

Wolfe, P., and G. B. Dantzig, 1962. "Linear Programming in a Markov Chain," Operations Research, 10, 5 (September-October), pp. 702-710.

*Wolfe, P., and L. Cutler, 1963. "Experiments in Linear Programming," in R. L. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, New York.

*Zoutendijk, G., 1960. Methods of Feasible Directions, Elsevier Publishing Company, Amsterdam.

Zschau, E. V. W., 1967. "A Primal Decomposition Algorithm for Linear Programming," Graduate School of Business Working Paper No. 91 (January), Stanford University, Stanford.

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY	2a. REPORT SECURITY CLASSIFICATION
THE RAND CORPORATION	UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

ELEMENTS OF LARGE-SCALE MATHEMATICAL PROGRAMMING

4. AUTHOR(S) (Last name, first name, initial)

Geoffrion, A. M.

5. REPORT DATE	6a. TOTAL NO. OF PAGES	6b. NO. OF REFS.
November 1969	90	---

7. CONTRACT OR GRANT NO.	8. ORIGINATOR'S REPORT NO.
F44620-67-C-0045	R-481-PR

9a. AVAILABILITY/LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	United States Air Force Project RAND

10. ABSTRACT

A unifying framework of concepts central to the optimization of large structured systems is developed and used in the organization of the literature. The principal focus is on that portion of the literature concerned with developing new solution techniques for various problem structures. The basic concepts are divided in two groups, (1) problem manipulations, in which a given problem is restated in an alternative form more amenable to solution, and (2) solution strategies which reduce an optimization problem to a related sequence of simpler problems that can be solved by specialized methods. By assembling elements of problem manipulation and solution strategies in various patterns, one can derive the essential aspects of most large-scale programming algorithms and even design new ones. Some of the literature is grouped according to five key patterns, and illustrative papers from each are discussed in detail. Some of the concepts presented in this study are considered in more technical detail in RM-5829.

11. KEY WORDS

Optimization
Mathematical programming
Problem solving